高等学校计算机应用规划教材

计算机英语

（第4版）

◆ 计算机系统及其组成
◆ 计算机体系结构
◆ 算法与数据结构
◆ 程序设计与语言
◆ 操作系统与应用软件
◆ 数据库系统、软件工程
◆ 面向对象技术
◆ 计算机网络与通信
◆ 信息安全、信息系统
◆ 企业资源规划
◆ 供应链与客户关系管理
◆ 电子商务
◆ 人工智能、大数据、云计算



姜同强　主　编

清华大学出版社

高等学校计算机应用规划教材

计 算 机 英 语

(第 4 版)

姜同强　主　编

清华大学出版社

北　　京

内 容 简 介

直接阅读外文技术资料和文献是每位计算机技术人员和研究人员必须具备的一种能力。本书旨在使学生及专业人员能够熟悉并掌握计算机方面的基本专业英文词汇，了解科技英语的基本认知规律，并提高在计算机专业英文文献方面的阅读能力。

本书共分为21章，汇集了计算机技术各方面的内容，包括计算机硬件、软件、网络与通信、计算机应用技术及人工智能、大数据、云计算等。本书内容和专业词汇的涵盖面广，选择的文章具有代表性和新颖性，其中的阅读材料对近10年来产生的一些新技术进行了详细介绍，从而使教师在选择教学内容方面有较大的灵活性。

本书适合于计算机科学与技术专业、软件工程专业、信息管理与信息系统专业、电子商务专业，以及其他相关专业的本科生、研究生作为计算机专业英语课程的教材，对从事与计算机相关工作的专业技术人员提高计算机专业外文文献的阅读能力也有一定的帮助。

图书在版编目(CIP)数据

计算机英语 / 姜同强　主编. —4 版. —北京：清华大学出版社，2020.5
高等学校计算机应用规划教材
ISBN 978-7-302-53218-7

Ⅰ.①计…　Ⅱ.①姜…　Ⅲ.①电子计算机－英语－高等学校－教材　Ⅳ.①TP3

中国版本图书馆 CIP 数据核字(2019)第 129374 号

责任编辑：刘金喜
封面设计：孔祥峰
版式设计：思创景点
责任校对：牛艳敏
责任印制：杨　艳

出版发行：清华大学出版社
网　址：http://www.tup.com.cn，http://www.wqbook.com
地　址：北京清华大学学研大厦 A 座　　邮　编：100084
社 总 机：010-62770175　　邮　购：010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62773464
印 装 者：三河市铭诚印务有限公司
经　销：全国新华书店
开　本：185mm×260mm　　印　张：22　　字　数：535 千字
版　次：2004 年 6 月第 1 版　2020 年 5 月第 4 版　　印　次：2020 年 5 月第 1 次印刷
定　价：59.00 元

产品编号：083109-01

第4版前言

本书在第3版的基础上，增加了近几年发展迅猛的新技术内容，如人工智能、大数据与云计算等。

全书内容分为6个部分。第一部分讲述计算机硬件基础，包括第1章——计算机系统概述，第2章——计算机系统的组成，第3章——计算机体系结构。第二部分讲述计算机软件基础，包括第4章——算法与数据结构，第5章——程序设计与语言，第6章——操作系统。第三部分讲述计算机软件，包括第7章——应用软件，第8章——数据库系统概论，第9章——软件工程，第10章——面向对象技术。第四部分讲述计算机网络与通信，包括第11章——计算机网络概述，第12章——局域网、城域网和广域网，第13章——Internet，第14章——信息安全。第五部分讲述计算机应用技术，包括第15章——信息系统，第16章——企业资源规划，第17章——供应链管理，第18章——客户关系管理，第19章——电子商务。第六部分讲述计算机新技术，包括第20章——人工智能，第21章——大数据与云计算。

本书主要读者对象是计算机科学技术专业、软件工程专业、信息管理与信息系统专业、电子商务专业及其他相关专业的本科生、研究生，以及从事计算机相关工作的专业人员。

为便于教学，本书提供了配套的教师用书和PPT教学课件。教师用书可通过填写书后的“教师用书申请卡”免费申请，PPT教学课件可通过 http://www.tupwk.com.cn/downpage 下载。

本书由姜同强主编，苗天顺、王雯、赵守香、曹倩、颜文婧等老师参加了编写。

在本书的编写和出版过程中，清华大学出版社的同志付出了辛勤的劳动，在此表示感谢。此外，还要感谢编写团队中的每一位成员，这些成员具有不同的专业背景，没有他们高效率的通力合作，就不可能在短时间内完成这样一本工作量巨大的教材编写工作。

由于编者水平有限，加之时间仓促，不足之处在所难免，欢迎广大读者批评指正。

服务邮箱：476371891@qq.com。

编　者

2019年10月于北京

第 3 版前言

本书第 2 版发行后，依然深受广大计算机专业英语老师和学生的欢迎。在吸取各方面意见的基础上，本书第 3 版在第 2 版的基础上，将学生用书中每一章的 Grammatical Notes to the Text 和阅读材料放入了教师用书中，压缩了学生用书的篇幅。

本书的内容分为 5 篇。第 1 篇——计算机硬件基础，包括第 1 章——计算机系统概述，第 2 章——计算机系统的组成，第 3 章——计算机体系结构。第 2 篇——计算机软件基础，包括第 4 章——算法与数据结构，第 5 章——程序设计与语言，第 6 章——操作系统。第 3 篇——计算机软件，包括第 7 章——应用软件，第 8 章——数据库系统概论，第 9 章——软件工程，第 10 章——面向对象技术。第 4 篇——计算机网络与通信，包括第 11 章——计算机网络概述，第 12 章——局域网、城域网和广域网，第 13 章——Internet，第 14 章——信息安全。第 5 篇——计算机应用技术，包括第 15 章——信息系统，第 16 章——人工智能与专家系统，第 17 章——企业资源规划，第 18 章——供应链管理，第 19 章——客户关系管理，第 20 章——电子商务。

本书主要读者对象是计算机科学技术专业、软件工程专业、信息管理与信息系统专业、电子商务专业及其他相关专业的本科生、研究生，以及从事计算机相关工作的专业人员。

为便于教学，本书提供了配套的教师用书和 PPT 教学课件，教师用书可随主教材一起订购，也可免费获赠，具体信息请见书后的“《教师用书》需求信息反馈卡”。PPT 教学课件可通过 http://www.tupwk.com.cn/downpage 下载。本书文中右上角带有注解序号的句子的语法解释，请参见教师用书。

本书由姜同强主编，苗天顺任副主编。姜同强、苗天顺负责全书的统稿。参加编写的人员包括(按章节顺序排列)：王雯编写第 1 章～第 6 章，姜同强编写第 7 章～第 10 章及各章专业术语的解释，孔凡航、周亦鹏编写第 11 章～第 15 章，赵守香编写第 16 章～第 20 章。另外，姜同强负责各章中 Technical Notes to the Text 和 Technical Terms and Proper Names 的编写；苗天顺负责编写各章语法注释和部分课后练习，盖爽编写了部分阅读材料，曹倩负责在第 2 版的基础上整理和调整第 3 版的内容。

在本书的编写和出版过程中，清华大学出版社的同志为使本书尽快出版付出了辛勤劳动，在此表示感谢。另外，还要感谢我们编写团队中的每一位成员，这些成员具有不同的专业背景，没有他们高效率的通力合作，就不可能在短时间内完成这样一本工作量巨大的教材编写。

由于编者水平有限，加之时间仓促，本书中出现的错误在所难免，欢迎广大读者批评指正。

服务邮箱：wkservice@vip.163.com。

编　者

2012 年 11 月于北京

第 2 版前言

本书第 1 版发行后，深受广大计算机专业英语老师和学生的欢迎，好评如潮。甚至参加全国计算机技术与软件专业技术资格(水平)考试的考生都将本书视为应试必读教材之一。短短 4 年时间已经多次印刷，印刷量突破几万册。有的老师在来信中说："我们在教学过程中多次使用清华大学出版社出版的《计算机英语》，效果很好。该书内容非常丰富，为教师的教学提供了极大的灵活性；该书提供配套的教师用书，为教师的备课提供了极大的方便。大多数此类教材中都包含课文的中文翻译，这样做既不利于提高学生学习计算机英语的效果，也不利于教师的备课，而且使课文的信息量大大降低，而本书在编排上将课文与翻译分离开来，彻底解决了上述问题，学生的学习效果和教学效果得到了极大的提高。"

另外，学生们反映，该书中专业术语的解释部分很受欢迎，在某种程度上起到了专业词典的作用，用起来很方便，而且专业术语的解释很详细，既提高了计算机英语的阅读能力，又学到了很多新的知识，可谓一石二鸟。

在吸取各方面意见的基础上，本书第 2 版针对第 1 版的内容做了以下几个方面的调整。

(1) 将每一章的课文进一步精练，并适当调整内容，压缩了篇幅。调整比较大的内容如下。

- 将第 1 版教材中的第 8 章(文件及文件处理)、第 15 章(多媒体技术)、第 19 章(数字图像处理)和第 24 章(计算机辅助设计、计算机辅助制造、计算机集成制造系统)删除。
- 将第 1 版教材中的第 12 章(计算机网络导论)和第 13 章(OSI 和 TCP/IP 参考模型)合并为一章，第 14 章(局域网和城域网)和第 15 章(广域网)合并为一章。
- 将第 1 版教材中的第 22 章(企业资源规划)分解为 3 章，分别是企业资源规划、供应链管理和客户关系管理。

(2) 每一章课文前增加了"Pre-reading Questions"内容以方便学生预习。课文后增加了以下几部分内容：Grammatical Notes to the Text，Words Bank to the Text(包括三部分内容，其中新增了 Useful New Words 和 Useful Phrases and Expressions 两部分内容)。

(3) 进一步丰富和规范了课后练习。

(4) 精练了课后的阅读材料，并进行了适当的调整和压缩。

(5) 为讲授此课的教师制作了配套的电子课件。

本书与同类教材相比，有如下几个方面的区别。

- 编写本书的教师队伍是由以下三个方面的人员构成的：计算机相关专业的教师、从事多年计算机专业英语教学的一线教师、从事多年普通英语教学的一线教师。
- 从教材的结构和内容编排来看，有其独到之处：既有专业词汇的正规解释，又包括了一些常见的语法现象的解释。从事本课程教学的教师无论是从事计算机专业还是英语专业，本书都为他们提供了极大的方便。
- 国内的大多数计算机英语教材在内容选取上都有所侧重：有的教材偏重于硬件；有的教材偏重于软件；还有的教材是信息电子类的，即侧重于通信电子方面。另外，有的教材侧重于理论，如数据结构、离散数学；而有的教材则侧重于应用，如软件工程、

数据库开发。本书的编写改变了这种状况，在理论和应用上，在硬件、软件、网络、应用等方面均有所体现。

本书的特色包括如下几个方面。

- 内容丰富，灵活性强。本书的内容非常丰富，涵盖了计算机科学技术专业及其相关专业的一些主要课程内容，包括计算机硬件、软件、网络与通信、计算机应用等，为不同专业教师的教学提供了可选性和极大的灵活性。
- 实用性和专业性相结合。本书的选材在保持“原汁原味”的同时使学习者更能接触到计算机英语的真实语境和主流思想，虽然有一定的难度，但非常实用和专业。另外，选材与我国大学本科专业教学计划中的专业课程有很好的对应关系。
- 重视教师的教学效果和学生的学习效果。大多数此类教材中都包含课文的中文翻译，这样做既不利于提高学生的学习效果，也不利于教师的备课，而且使课文的信息量大大降低。而本书在编排上将课文与翻译分离开来，彻底解决了上述问题，学生的学习效果和教师的教学效果得到了极大的提高。
- 附赠教师用书：凡选用本书作为教材的教师，均可免费获赠《计算机英语 • 教师用书(第 2 版)》。具体方法请参见书后的“《教师用书》需求信息反馈卡”。

本书从内容上可分为 5 篇。第 1 篇——计算机硬件基础，包括第 1 章——计算机系统概述，第 2 章——计算机系统的组成，第 3 章——计算机体系结构。第 2 篇——计算机软件基础，包括第 4 章——算法与数据结构，第 5 章——程序设计与语言，第 6 章——操作系统。第 3 篇——计算机软件，包括第 7 章——应用软件，第 8 章——数据库系统概论，第 9 章——软件工程，第 10 章——面向对象技术。第 4 篇——计算机网络与通信，包括第 11 章——计算机网络概述，第 12 章——局域网、城域网和广域网，第 13 章——Internet，第 14 章——信息安全。第 5 篇——计算机应用技术，包括第 15 章——信息系统，第 16 章——人工智能与专家系统，第 17 章——企业资源规划，第 18 章——供应链管理，第 19 章——客户关系管理，第 20 章——电子商务。

本书主要读者对象是计算机科学技术专业、软件工程专业、信息管理与信息系统专业、电子商务专业及其他相关专业的本科生、研究生，以及从事计算机相关工作的专业人员。

本书由姜同强主编，苗天顺任副主编。姜同强、苗天顺负责全书的统稿。参加编写的人员包括(按章节顺序排列)：王雯编写第 1 章～第 6 章，姜同强编写第 7 章～第 10 章及各章专业术语的解释，孔凡航、周亦鹏编写第 11 章～第 15 章，赵守香编写第 16 章～第 20 章。另外，姜同强负责各章中 Technical Notes to the Text 和 Technical Terms and Proper Names 的编写；苗天顺负责编写各章语法注释和部分课后练习，盖爽编写了部分阅读材料。

在本书的编写和出版过程中，清华大学出版社的同志为使本书尽快出版付出了辛勤劳动，在此表示感谢。另外，还要感谢我们编写团队中的每一位成员，这些成员具有不同的专业背景，没有他们高效率的通力合作，就不可能在短时间内完成这样一本工作量巨大的教材编写。

由于编者水平有限，加之时间仓促，本书中出现的错误在所难免，欢迎广大读者批评指正。

作者的联系方式：jtongqiang@yahoo.com.cn。

服务邮箱：wkservice@vip.163.com。

编　者

2008 年 9 月于北京

第 1 版前言

随着时代的发展，计算机与网络技术已渗透到人们工作和生活的各个方面。计算机英语也随之独立成为一门专业英语，并在计算机应用中作为人机之间交流的媒介。

一个计算机方面的人才除了要掌握计算机理论和技能以外，更重要的是具备快速获取新的计算机方面知识的能力。而计算机英语(尤其是阅读能力)则是体现这种能力的一个重要方面。本书正是在这样的指导思想下编写的。

1. 编写目的

- 使学生熟悉并掌握计算机方面的基本专业英文词汇。
- 提高学生的计算机专业英文文献的阅读能力。

2. 本书特点

- 系统性：本书涵盖了计算机技术各个方面的内容，包括计算机硬件、软件、网络与通信、计算机应用等。
- 新颖性：本书反映了 20 世纪 90 年代到 21 世纪初的最新技术。
- 代表性：本书选择的文章在内容上具有一定的代表性，基本体现了计算机硬件、计算机软件、网络与通信和计算机应用方面的典型技术。
- 广泛性：本书专业词汇的涵盖面广。
- 附赠配套教材：凡选用本书作为教材的教师，均可免费获赠《计算机英语 ·教师用书》。具体方法请见书后的“《教师用书》需求信息反馈卡”。

3. 本书的结构及内容

本书从内容上可分为 5 篇。第 1 篇——计算机硬件基础，包括第 1 章——计算机系统概述，第 2 章——计算机系统的组成，第 3 章——计算机体系结构。第 2 篇——计算机软件基础，包括第 4 章——算法与数据结构，第 5 章——程序设计与语言，第 6 章——操作系统，第 7 章——应用软件，第 8 章——文件和文件处理，第 9 章——数据库系统概论，第 10 章——软件工程，第 11 章——面向对象技术。第 3 篇—— 计算机网络与通信，包括第 12 章—— 计算机网络概述，第 13 章——OSI 参考模型和 TCP/IP 参考模型，第 14 章——局域网和城域网，第 15 章——广域网，第 16 章——Internet，第 17 章——网络安全。第 4 篇——其他计算机技术，包括第 18 章——多媒体技术，第 19 章——数字图像处理，第 20 章——人工智能与专家系统。第 5 篇——计算机应用，包括第 21 章——计算机信息系统，第 22 章——企业资源规划，第 23 章——电子商务，第 24 章——CAD/CAM/CIMS。

每章除了正文外，还列举出本章的专业词汇对照表、重点词汇的详细说明，正文后还附有练习题，可作为对学生学习情况的检测。每章最后的阅读材料是对正文内容的补充，反映

了最新的技术，可作为学生课后阅读的内容，加深对正文内容的理解。

4. 读者对象

本书主要读者对象是计算机专业及相关专业的高职、高专、本科学生和从事计算机相关工作的专业人员。

本书由姜同强主编。参加编写的人员包括(按章节顺序排列)：王雯、罗代洪编写第 1 章、第 2 章、第 3 章、第 4 章、第 5 章和第 6 章，姜同强、杨冰编写第 7 章、第 8 章、第 9 章、第 10 章和第 11 章，孔凡航、吕燕编写第 12 章、第 13 章、第 14 章、第 15 章、第 16 章、第 17 章和第 18 章，赵守香编写第 19 章、第 20 章、第 21 章、第 22 章、第 23 章和第 24 章。王振玲对全书内容进行了审校。

在本书的编写和出版过程中，清华大学出版社的同志为使本书尽快出版付出了辛勤劳动，在此表示感谢。

由于编者水平有限，加之时间仓促，对于本书中出现的错误，欢迎广大读者批评指正。

编　者

2004 年 6 月

目　　录

Chapter 1　Computer System Overview ······ 1
1.1　Digital Computer ······ 1
1.2　Data Types ······ 2
1.3　The Evolution of Computers ······ 3
1.4　Types of Computers ······ 4
Technical Notes to the Text ······ 5
Word Bank to the Text ······ 7
Exercises ······ 10

Chapter 2　Computer System Organization ······ 13
2.1　Computer Organization Introduction ······ 13
2.2　System Buses ······ 14
2.2.1　Address Bus ······ 14
2.2.2　Data Bus ······ 14
2.2.3　Control Bus ······ 14
2.3　CPU Organization ······ 15
2.3.1　Register Set ······ 15
2.3.2　Arithmetic Logic Unit ······ 15
2.3.3　Control Unit ······ 15
2.4　Memory Subsystem Organization ······ 16
2.4.1　Types of Memory ······ 16
2.4.2　Memory Hierarchy ······ 18
2.5　I/O Subsystem Organization ······ 19
2.5.1　I/O Devices ······ 19
2.5.2　I/O Interface ······ 20
2.5.3　Modes of Transfer ······ 21
Technical Notes to the Text ······ 23
Word Bank to the Text ······ 25
Exercises ······ 28

Chapter 3　Computer System Architecture ······ 32
3.1　Parallel Processing ······ 32
3.2　Pipelining ······ 33
3.3　Vector Processing ······ 35
3.4　Reduced Instruction Set Computer (RISC) ······ 36
Technical Notes to the Text ······ 38
Word Bank to the Text ······ 39
Exercises ······ 41

Chapter 4　Algorithms and Data Structure ······ 44
4.1　Algorithms ······ 44
4.2　Data Structure ······ 49
Technical Notes to the Text ······ 55
Word Bank to the Text ······ 56
Exercises ······ 59

Chapter 5　Programming and Languages ······ 63
5.1　The Procedure of Programming ······ 63
5.2　The Development of Programming Languages ······ 64
5.3　Compiling and Assembling Programs ······ 66
5.4　Object-Oriented Programming (OOP) ······ 67
5.5　Visual Programming ······ 72
5.6　Internet Programming ······ 73
Technical Notes to the Text ······ 74
Word Bank to the Text ······ 76

Exercises ······ 79

Chapter 6　Operating System ······ 82
6.1　Operating System Overview ······ 82
6.2　Operating System Platform ······ 88
Technical Notes to the Text ······ 91
Word Bank to the Text ······ 92
Exercises ······ 95

Chapter 7　Applications Software ······ 98
7.1　Applications Software Tools ······ 98
7.2　Common Features of Applications Software ······ 99
7.3　Productivity Software Tools ······ 100
7.3.1　Word Processing Software ······ 101
7.3.2　Spreadsheet Software ······ 102
7.3.3　Presentation Graphics ······ 103
7.3.4　Groupware ······ 103
7.3.5　Desktop Accessories ······ 104
7.3.6　Web Browsers ······ 105
Technical Notes to the Text ······ 105
Word Bank to the Text ······ 106
Exercises ······ 109

Chapter 8　An Introduction to Database Systems ······ 112
8.1　Purpose of Database Systems ······ 112
8.2　View of Data ······ 114
8.2.1　Levels of Data Abstraction ······ 114
8.2.2　Instances and Schemas ······ 115
8.2.3　Data Independence ······ 116
8.3　Data Models ······ 116
8.3.1　Object-Based Logical Models ······ 117
8.3.2　Record-Based Logical Models ······ 118
8.3.3　Physical Data Models ······ 120
8.4　Database Languages ······ 120
8.4.1　Data Definition Language (DDL) ······ 120
8.4.2　Data Manipulation Language (DML) ······ 121
8.4.3　The Structured Query Language (SQL) ······ 121
8.5　Transaction Management ······ 122
8.6　Database Administrator ······ 123
Technical Notes to the Text ······ 124
Word Bank to the Text ······ 126
Exercises ······ 128

Chapter 9　Software Engineering ······ 132
9.1　What Is Software Engineering? ······ 132
9.2　Key Issues of Software Engineering ······ 132
9.3　Software Process ······ 136
9.4　Computer-Aided Software Engineering (CASE) ······ 141
Technical Notes to the Text ······ 142
Word Bank to the Text ······ 144
Exercises ······ 147

Chapter 10　Object-Oriented Technology ······ 150
10.1　A Brief Overview of Object Technology ······ 150
10.2　What Is OO—— System Concepts for Object Modeling ······ 150
10.3　The OO Development Process ······ 156
10.4　Unified Modeling Language (UML) ······ 160
Technical Notes to the Text ······ 162
Word Bank to the Text ······ 164
Exercises ······ 167

Chapter 11　Introduction to Computer Networks ······ 170
11.1　Data Communications ······ 170
11.1.1　Signals ······ 170
11.1.2　Encoding ······ 171
11.1.3　Transmission Mode ······ 172

11.2 Introduction to Computer Networks ··· 172
11.3 Applications of Computer Networks ··· 173
11.4 Categories of Networks ··· 174
11.4.1 Configurations ··· 174
11.4.2 Strategies ··· 175
11.4.3 LANs, MANs, and WANs ··· 177
11.4.4 Intranets and Extranets ··· 178
11.5 OSI and TCP/IP Reference Model ··· 179
11.5.1 OSI Reference Model ··· 179
11.5.2 TCP/IP Reference Model ··· 181
Technical Notes to the Text ··· 182
Word Bank to the Text ··· 183
Exercises ··· 186

Chapter 12 LAN, MAN & WAN ··· 190
12.1 Local Area Networks (LANs) ··· 190
12.1.1 Ethernet (802.3) ··· 190
12.1.2 Token Bus (802.4) ··· 192
12.1.3 Token Ring (802.5) ··· 192
12.1.4 Fiber Distributed Data Interface (FDDI) ··· 194
12.1.5 Comparison ··· 195
12.2 Metropolitan Area Networks (IEEE 802.6) ··· 196
12.3 Wide Area Networks ··· 196
12.3.1 Introduction ··· 196
12.3.2 Narrowband ISDN ··· 197
12.3.3 X.25 Network ··· 198
12.3.4 Frame Relay ··· 198
12.3.5 Broadband ISDN and ATM ··· 199
Technical Notes to the Text ··· 199
Word Bank to the Text ··· 200
Exercises ··· 203

Chapter 13 Internet ··· 206
13.1 Introduction ··· 206
13.2 Technology of Internet ··· 207
13.2.1 Internet Address ··· 207
13.2.2 DNS (Domain Name System) ··· 208
13.2.3 HTTP and SMTP ··· 210
13.3 Services Provided by the Internet ··· 211
13.3.1 WWW (World Wide Web) ··· 211
13.3.2 E-mail ··· 211
13.3.3 FTP (File Transfer Protocol) ··· 212
13.3.4 Telnet ··· 213
13.4 Networking Devices ··· 213
13.4.1 Hub and Repeater ··· 213
13.4.2 Bridge and Switch ··· 213
13.4.3 Router ··· 214
13.5 Access to Internet ··· 215
13.5.1 Dial-up ··· 215
13.5.2 DSL ··· 215
Technical Notes to the Text ··· 216
Word Bank to the Text ··· 216
Exercises ··· 219

Chapter 14 Information Security ··· 222
14.1 A Brief Overview of Information Security Concepts ··· 222
14.2 Information Security Technology ··· 225
14.2.1 Information Authentication Technology ··· 225
14.2.2 Encryption Technology ··· 226
14.2.3 Database Security ··· 229
14.3 Computer Virus ··· 230
14.3.1 Virus Behavior ··· 230
14.3.2 Types of Viruses ··· 230

14.3.3 Virus Prevention and Detection ········ 231
14.4 Firewall ········ 232
14.4.1 Concept of Firewall ········ 232
14.4.2 Types of Firewall ········ 233
14.4.3 Firewall Implementation ········ 233
14.5 Standards of Information Security ········ 235
14.6 E-Commerce Security ········ 236
14.6.1 Significance of Corporate Information Security ········ 236
14.6.2 Current Processes and Tools for Implementing E-Business Security ········ 236
Technical Notes to the Text ········ 237
Word Bank to the Text ········ 238
Exercises ········ 241

Chapter 15 Information System ········ 245
15.1 Major Types of Information System in Organization ········ 245
15.2 Transaction Processing System (TPS) ········ 246
15.3 Management Information System (MIS) ········ 247
15.3.1 A Definition of MIS ········ 247
15.3.2 A MIS Model ········ 248
15.3.3 Organizational Information System ········ 248
15.4 Decision Support System (DSS) ········ 249
15.4.1 The Concept of DSS ········ 249
15.4.2 DSS Objectives ········ 251
15.4.3 A DSS Model ········ 251
Technical Notes to the Text ········ 252
Word Bank to the Text ········ 254
Exercises ········ 256

Chapter 16 Enterprise Resource Planning ········ 259
16.1 Enterprise System ········ 259
16.2 Enterprise Resource Planning System ········ 260
16.3 The Evolution of Enterprise Resource Planning ········ 262
Technical Notes to the Text ········ 266
Word Bank to the Text ········ 267
Exercises ········ 268

Chapter 17 Supply Chain Management ········ 272
17.1 What Is a Supply Chain ········ 272
17.2 The Objectives of a Supply Chain ········ 273
17.3 Decision Phases in a Supply Chain ········ 274
17.4 The Importance of Supply Chain Flows ········ 276
Technical Notes to the Text ········ 277
Word Bank to the Text ········ 278
Exercises ········ 279

Chapter 18 Customer Relationship Management ········ 283
18.1 Customer Relationship Management Concepts ········ 283
18.2 Four Types of Customers ········ 284
18.2.1 Win Back or Save ········ 284
18.2.2 Prospecting ········ 284
18.2.3 Loyalty ········ 285
18.2.4 Cross-Sell/Up-Sell ········ 285
18.3 Customer Relationship Management (CRM) Software ········ 286
18.3.1 Sales Force Automation (SFA) ········ 286
18.3.2 Customer Service ········ 286

18.3.3 Marketing ··· 287
18.4 Operational and Analytical CRM ··· 287
Technical Notes to the Text ··· 288
Word Bank to the Text ··· 289
Exercises ··· 291

Chapter 19 E-Business and E-Commerce ··· 295
19.1 E-Business ··· 295
19.2 E-Commerce ··· 299
Technical Notes to the Text ··· 300
Word Bank to the Text ··· 301
Exercises ··· 303

Chapter 20 Artificial Intelligence ··· 307
20.1 History of AI ··· 307
20.2 Basics of AI ··· 308
20.3 Applications of Artificial Intelligence in Machine Learning ··· 309
Technical Notes to the Text ··· 312
Word Bank to the Text ··· 314
Exercises ··· 316

Chapter 21 Big Data and Cloud Computing ··· 320
21.1 Background ··· 320
21.2 Big Data ··· 321
21.3 Cloud Computing ··· 324
Technical Notes to the Text ··· 326
Word Bank to the Text ··· 327
Exercises ··· 329

Chapter 1
Computer System Overview

Pre-reading Questions

1. What is a digital computer?

2. Are there any differences between the binary number system and the common decimal number system?

3. How many types of computers do you know? Name at least four of them.

Digital computer is also called electronic computer or computer. Computers surround us. It's hard to find a field in which computers are not being used. In this chapter, we will introduce digital computer, data types, the evolution of computers, and types of computers.

1.1 Digital Computer

The digital computer is a digital system that performs various computational tasks. The word "digital" implies that the information in the computer is represented by variables that take a limited number of discrete values. These values are processed internally by components that can maintain a limited number of discrete states. The decimal digits 0, 1, 2 … 9, for example, provide 10 discrete values. The first electronic digital computers, developed in the late 1940s, were used primarily for numerical computations. In this case, the discrete elements are the digits. From this application, the term "digital computer" has emerged. In practice, digital computer functions more reliably if only two states are used. Because of the physical restriction of components, and because human logic tends to be binary, digital components that are constrained to take discrete values are further constrained to take only two values and are said to be binary.

Digital computers use the binary number system, which has two digits: 0 and 1. A binary digit is called bit. Information is represented in digital computer in groups of bits. By using various coding techniques, groups of bits can be made to represent not only binary numbers but also other discrete symbols, such as decimal digits or letters of the alphabet. For example, American Standard Code for Information Interchange (ASCII) originally used 7 bits to form a character. By judicious use of binary arrangements and by using various coding techniques, the groups of bits are used to develop complete sets of instructions for performing various types of computations. In contrast to the common decimal numbers that employ the base 10 system, binary numbers use a base 2 system with two digits: 0 and 1. The decimal equivalent of a binary number can be found by expanding it into a power series with a base of 2.

A computer system consists of hardware system and software system. The hardware system is the physical equipment that you can see and touch, such as the disks and the screen. The software system is the intangible “control” that governs the computer; it is the total of all the programs that can be run on the computer. A program is a list of instructions. Programs tell the hardware what to do. The hardware of the computer is usually divided into three major parts: input and output devices (I/O devices), a central processing unit (CPU), and memory. They will be described in more detail in Chapter 2. Software can be classified according to its purpose. Application software is designed to accomplish real-world tasks in fields such as accounting, entertainment, and engineering. If you’ve ever played a video game or typed a paper on a word processor, you’ve already had some experience with application software programs. System software, on the other hand, controls the computer system itself. System software includes not only the complex programs used by technicians to create application software in the first place but also the organizational programs needed to start up the computer and govern its use of other programs. They will be described in more detail in Chapter 2 and Chapter 6.

1.2 Data Types

Binary information in digital computers is stored in memory or processor registers. Registers contain either data or control information. Control information is a bit or a group of bits used to specify the sequence of command signals needed for manipulation of the data in other registers. Data are numbers and other binary-code information that are operated on to achieve required computational results. Now we present the most common types of data found in digital computers and show how the various data types are represented in binary-code form in computer registers.

The data types found in the registers of digital computers may be classified into the following categories:

- Numeric data can often be represented as integers. In unsigned integers, an n-bit value can range from 0 to 2n–1. An n-bit signed integer can have any value between–2n–1 and 2n–1–l, inclusive. Both formats can be used in arithmetic algorithms. Some numeric data cannot be represented as integers. These values, which typically include fractional portions, are represented in floating point format in computers. A computer may have special registers and instructions that are exclusive for floating point data.
- The Boolean values TRUE and FALSE are used often enough to warrant having their own data type, Boolean, and assembly language instructions. Typically, a data value is set to zero to represent FALSE and any nonzero value for TRUE. Boolean assembly language instructions can perform logical operations on these values. Unlike logical instructions, which generate one result per bit of the operands, Boolean instructions generate only one result. To illustrate the difference, consider the case in which A=0000 0010 and B=0000 0001. The logical AND of these binary values produces the result 0000 0000. However, if

they are Boolean values, A and B are both TRUE, since they are both nonzero. Their Boolean AND must produce a result of TRUE, represented by a nonzero value.

- Computers must also deal with character data. The characters are stored as binary values encoded using ASCII, EBCDIC, UNICODE, or some other character encoding standards. Rather than arithmetically or logically manipulating characters, a computer may concatenate strings of characters, replace some characters with others, or otherwise manipulate character strings. Some assembly language instruction sets include instructions to directly manipulate character data. Others use routines constructed from other instructions to achieve the same result.

1.3 The Evolution of Computers

The first large-scale electronic computer was the Electronic Numerical Integrator and Computer (ENIAC), which became operational in 1946. From then on, computers have developed through four generations, or stages, each one characterized by smaller size and less expense than its predecessor.

1. First Generation (1944-1958)

In the earliest general-purpose computer, most input and output media were punched cards and magnetic tape. Main memory was almost exclusively made up of hundreds of vacuum tubes—although one computer used a magnetic drum for main memory. These computers were somewhat unreliable because the vacuum tubes failed frequently. They were also slower than any microcomputer used today, produced a tremendous amount of heat, and were very large. They could run only one program at a time.

2. Second Generation (1959-1963)

By the early 1960s, transistors and some other solid-state devices that were much smaller than vacuum tubes had been used for much of the computer. Magnetic cores, which looked like very small metal washers strung together by wires that carried electricity, became the most widely used type of main memory. Removable magnetic disk packs, stacks of disks connected by a common spindle, were introduced as storage devices. Second-generation machines tended to be smaller, more reliable, and significantly faster than first-generation computers.

3. Third Generation (1964-1970)

In the third period, the integrated circuit (IC) —a complete electronic circuit that packages transistors and other electronic components on a small silicon chip — replaced traditional transistorized circuitry. Integrated circuits are cost-effective because individual components don’t need to be wired directly to the computer’s system board.

The use of magnetic disks for secondary data storage became widespread, and computers

began to support such capabilities as multiprogramming (processing several programs simultaneously) and timesharing (people using the same computer simultaneously). Minicomputers had been widely used by the early l970s and were taking some of the business away from the established mainframe market. Processing that formerly required the processing power of a mainframe could now be done on a minicomputer.

4. Fourth Generation (1971-Now)

Large-scale integrated (LSI) andvery-large-scale integrated (VLSI) circuits were developed, which contained hundreds to millions of transistors on a tiny chip. In 1971, Ted Hoff of Intel developed the microprocessor, which packaged an entire CPU, complete with memory, logic, and control circuits, on a single chip. The microprocessor and VLSI circuit technology caused radical changes in computers—in their size, appearance, cost, availability and capability, and they started the process of miniaturization—the development of smaller and smaller computers.

Also during this time, a computer's main memory capacity increased, and its cost decreased, which directly affected the types and usefulness of software that could be used. Software applications like word processing, electronic spreadsheets, database management programs, painting and drawing programs, desktop publishing, and so forth became commercially available, giving more people reasons to use a computer.

1.4　Types of Computers

Computers are usually classified into four broad categories: microcomputers, minicomputers, mainframe computers and supercomputers. It's hard to give a precise definition to each type because computer speeds and storage capacities change rapidly. Nevertheless, the following definitions will suffice.

1. Microcomputers

Microcomputers, also called personal computers (PC), are small computers that can fit next to a desk or on a desktop, or can be carried around. Microcomputers are either used as stand-alone computers or connected to a network, such as a local area network (LAN). A local area network connects, usually by special cable, a group of desktop personal computers and peripheral devices in an office or a building.

2. Minicomputers

Minicomputers are designed to support many time-sharing terminals at once. Minicomputers operate faster and are more expensive than microcomputers. Often, a minicomputer satisfies the general-purpose computing needs of a department or small business. Other minicomputers are dedicated to specific applications. For example, a minicomputer may be used to control an assembly line in a factory, recode data in a research laboratory, or help programmers develop

programs for other computers.

3. Mainframe computers

Mainframe computers are larger, faster, and more expensive than minicomputers. Mainframe computers also have many processors. They are found in banks, insurance companies, airlines, large corporations, and government organizations. Mainframes often serve hundreds of timesharing users at once. Mainframes are ideal for problems requiring extensive mathematical calculations or for sharing large volumes of information among many people.

4. Supercomputers

First developed in the 1970s, supercomputers are the fastest and highest-capacity computers. Their cost ranges from several hundreds of thousands to millions of dollars. They may occupy special air-conditioned rooms and are often used for research. Among their uses are worldwide weather forecasting and analysis of weather phenomena, oil exploration, aircraft design, evaluation of aging nuclear weapons systems, and mathematical research. Unlike microcomputers, which generally have only one central processing unit, supercomputers have hundreds to thousands of processors and can perform trillions of calculations per second.

Technical Notes to the Text

1. digital computer，数字计算机。数字计算机是指能执行数学计算和逻辑运算的计算机，其值通常用二进制数字表示。

2. binary，二进制。二进制是以 2 为基数的一种计数系统。常见的用法有 binary system(二进制数字系统)、binary code(二进制码)。

3. bit，(二进制)位，比特，同 binary digit。在二进制计数系统中，bit 代表数字 0 和 1 中的任何一个。其是存储设备中的最小信息容量单位，如 bits per second，缩写为 b/s，指每秒钟传输的位数。

4. ASCII (American standard code for information interchange)，美国信息交换标准代码。其是一种在不同厂家生产的设备之间为信息交换而定义编码的标准。这种编码使用由 7 位编码字符组成的编码字符集，用于数据处理系统、数据通信系统和有关设备之间的信息交换。ASCII 字符集由 128 个代码组成，其中 96 个是大小写字母、数字和符号，32 个为控制符。

5. coding，编码(方法，技术)，程序。常见的用法有 coding techniques(编码技术)、binary coding(二进制编码)。

6. input and output device (I/O device)，输入/输出设备。输入设备是可将用户所输入的程序、数据、操作命令等信息变换成计算机能接受的二进制形式的信息并输入内存中，以便计算机进行处理的设备，如键盘、鼠标等。输出设备是可将计算机的处理结果转换成人或其他设备可识别和接受的形式并将其表现出来的设备，如显示器、打印机等。

7. central processing unit (CPU)，中央处理器。它是由控制单元(CU)、算术 / 逻辑运算单元(ALU)和存储单元(包括寄存器和高速缓存)三个部分组成的。三个部分通过互连机构的连接，互相配合，共同完成对指令信息和数据信息的分析、判断、运算，从而控制计算机各部件协调工作。

8. register，寄存器。寄存器是一种存储装置，具有规定的存储容量，如一个位、一个字节或一个计算机字，通常有特定的用途。

9. unsigned integer，无符号整数。常见的表达方式有 signed integer(带符号整数)、signless integer(无符号整数，正整数)。

10. data type，数据类型。数据类型是指可由程序设计语言直接说明的数据的结构特性、特点和特征，如 FORTRAN 语言中的整数和实数。

11. floating point，浮点(数)，可缩写成 FLP 或 FP。其是一种表示数的形式，数目可表示为一个数乘以基数的幂次，如十进制数 397 可写成 3.97×10^2 或 0.397×10^3。

12. Boolean，布尔值，布尔。布尔值有真(用 True 表示)和假(用 False 表示)两个值。

13. assembly language，汇编语言。其是一种接近于二进制机器指令的程序语言。

14. operand，操作数，运算数，运算对象；运算域。

15. UNICODE，统一的字符编码标准，采用双字节对字符进行编码。

16. routine，例行程序，例程。其是用来执行一项特定且有限的任务的一套程序指令。

17. ENIAC (electronic numerical integrator and calculator[computer])，电子数字积分计算机，是世界上第一台通用计算机的名称，1946 年由美国制造。

18. punched card，穿孔卡。在字(词)处理技术中，其上用穿孔行来代表文本及程序指令。

19. magnetic tape，磁带。其是一种具有可磁化表面层的带状物，以磁记录方式存储数据。

20. vacuum tube，真空管，可缩写为 VT。其是一种内部空气全部或部分抽空的电子管，从而使电子在不受或少受空气分子的干扰下运动。

21. magnetic drum，磁鼓。具有磁性表面涂层的直圆柱体，数据以磁记录方式存储在它的磁性表面上。

22. main memory，主存储器。计算机中最主要的存储设备。常见的类似表达方式有以下几种。

- auxiliary memory (辅助存储器，辅助性记忆装置)
- buffer memory (缓冲存储器，超高速缓冲存储器)
- dynamic random access memory (动态随机存取存储器)
- dynamic memory (动态存储器)
- external memory (外存储器，外部记忆装置)
- hypothetical memory (虚拟存储器)

23. magnetic core，磁芯。通常为环状的、用作存储器的磁性材料。

24. integrated circuit (IC)，集成电路。一种微型的材料片或材料块，上面集成了复杂的电子元件和它们的连接线路。

25. silicon chip，硅片。为大量集成电路提供半导体基片的一种硅(圆)片。

26. multiprogramming，多道程序设计(技术，方法)；多道程序，可缩写成 MP。

27. minicomputer，小型计算机。一种比微型计算机内存容量大且运行速度高的小型计算机，通常可装入一个机柜。

28. mainframe，主机，大型计算机。

- 主机。计算机除了外部和分离的装置以外的中心运作部件。
- 大型计算机。大型的、强大的计算机，经常用作一些相互连接的终端服务器。

29. microprocessor，微处理器。其是一个芯片上包含有计算机的全部中央处理组件的集成电路。

30. personal computer (PC)，个人计算机。供个人用的微型计算机，可在办公室、家里或学校使用。

31. supercomputer，巨型计算机。其是一种主计算机，在一个特定时间内是最大、最快或功率最高的计算机之一。

Word Bank to the Text

A. Useful new words

surround	*v*. 包围，环境
computational	*adj*. 计算上的
discrete	*adj*. 不连续的，离散的
constrain	*v*. 抑制，约束，束缚
decimal	*adj*. 小数的，十进位的
judicious	*adj*. 明智的，明断的
equivalent	*adj*. 相等的，相当的
intangible	*adj*. 难以明了的，无形的
specify	*v*. 指定，载明
manipulation	*n*. 处理，操作
algorithm	*n*. 算法
fractional	*adj*. 分数的，部分的
warrant	*v*. 保证，担保
concatenate	*v*. 使连接起来，连锁
spindle	*n*. 轴，杆，心轴
simultaneously	*adv*. 同时地
dedicated	*adj*. 专用的
variable	*n*. 变量，变数

B. Useful expressions

in this case/in that case	既然是这样/那样，假使这样/那样的话
emerge (from)	出现，出来，产生
in practice	在实践中，实际上
tend to	倾向于，很可能，常常会，往往会
not only…but also	不但……而且
such as	诸如，像……一样
in contrast to/with	与……形成对比，与……截然不同
consist of	由……构成/组成
be made up of	由……组成/构成
in detail	详细地，详尽地
according to	依据，根据，按照
in the first place	首先，第一(点)
start up	开动，启动，发动
in…form	以……的形式
range from…to …	相当于 vary from…to… (在一定范围内)从……到……变动， 从……到……变化
deal with	处理
rather than	(而)不是
replace…with/by…	以……代替
at a time	一次，每次
and so forth/and so on	等等，诸如此类
either…or	或者……或者，要么……要么
at once=at the same time	同时

C. Technical terms and proper names

digital computer	数字计算机
decimal digit	十进制数字
binary	二进制
bit	位，比特
ASCII	美国信息交换标准代码
computer system	计算机系统
hardware system	硬件系统
software system	软件系统
I/O device	输入/输出设备
central processing unit (CPU)	中央处理器

memory	存储器
application software	应用软件
video game	计算机游戏
system software	系统软件
register	寄存器
floating point data	浮点数据
Boolean	布尔值，布尔
character data	字符数据
punched cards	穿孔卡片
magnetic tape	磁带
main memory	主存储器
vacuum tube	电子管，真空管
magnetic drum	磁鼓
transistor	晶体管
solid-state device	固态器件
magnetic core	磁芯
integrated circuit (IC)	集成电路
silicon chip	硅芯片
multiprogramming	多道程序设计
timesharing	分时，分时技术
minicomputer	小型计算机
mainframe	大型计算机
large-scale integrated (LSI)	大规模集成
very-large-scale integrated (VLSI)	超大规模集成
word processing	文字处理
electronic spreadsheet	电子表格
database management program	数据库管理程序
desktop publishing	桌面印刷
personal computers (PC)	个人计算机
microcomputer	微型计算机，微机
storage capacity	存储容量
stand-alone computer	独立计算机
local area network (LAN)	局域网
peripheral device	外部设备，外设
assembly line	流水线，生产线
supercomputer	巨型计算机

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. __________ is designed to accomplish real-world tasks in some fields.
2. An 8-bit signed integer can have any value between _________ and _________.
3. _________ controls the computer and enables it to run the hardware and application software.
4. A computer system consists of _________ and _________.
5. List four types of computers:_________, _________, _________ and _________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The software system is the physical equipment that you can see and touch. (　　)
2. Typically, a data value is set to zero to represent FALSE and 1 value for TRUE. (　　)
3. In the earliest general-purpose computer, most input and output media were magnetic disks. (　　)
4. Supercomputers are the largest, fastest, and most expensive computer available. (　　)
5. A computer system consists of hardware system and software system. (　　)

Ⅲ. Match each of the following terms with the appropriate definition.

IC	CPU	bit	ASCII

1. _______ The processing unit at the heart of a computer.
2. _______ A unit of information conveyed by a single binary digit.
3. _______ A complete electronic circuit that packages transistors and other electronic components on a small silicon chip.
4. _______ A system for encoding characters as binary digits.

Ⅳ. Translate the following into Chinese.

1. coding technique　　2. application software　　3. floating point data
4. timesharing　　5. storage capacity

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

specify	equivalent	warrant
emerge	magnetic	simultaneously
dedicated	decimal	discrete

1. He ______ from the accident unharmed.

2. This material is ______ to be pure silk.

3. He changed his pounds for the ______ amount of dollars.

4. The regulations ______ that you may use a dictionary in the examination.

5. ______ mathematics has become popular in recent decades because of its applications to computer science.

6. The _________ 0.61 stands for 61 hundredths.

7. The electrically charged gas particles are affected by _________ forces.

8. Bessie has _________ her life to caring for others.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "increase" and fill in the sentences with the right word.

Example: A special very-high-speed memory called a cache is sometimes used to **increase** the **speed** of processing by making current programs and data available to the CPU at a rapid rate.

1. The aim of this course is to increase the awa______ of self-worth of our black children.

2. The plan is to increase the effi______ of the railway services and provide more choices for passengers.

3. When we are learning a new language, we have to take into account the increasing inter_____ of different cultures in society.

4. Their purposes in studying in the United States include increasing their com______ of English, finishing high school, and understanding the American way of life as much as possible.

5. This new teaching tool shows how to combine traditional classroom techniques with the use of video and this will increase the con_______ of teachers who may be unfamiliar with the medium.

6. He increased his sp______ to overtake the lorry.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

This chapter introduces digital computer, data types, the evolution of computers, and types of computers. __1__ is known to all, it's hard to find a field in __2__ computers are not being used. Digital computer, also called electronic computer or computer, is a digital system that __3__ various computational tasks. Digital computers use the __4__ number system, which has two digits: 0 and 1. By using various coding __5__, groups of bits can be made to represent not only binary numbers __6__ other discrete symbols, such as decimal digits or letters of the alphabet. A computer system consists of hardware system and software system. Programs tell the hardware what to do. __7__ software is designed to accomplish real-world tasks in fields such as

accounting, entertainment, and engineering. Computers are usually ___8___ into four broad categories: microcomputers, minicomputers, mainframe computers, and supercomputers. It's hard to give a ___9___ definition to each type because computer speeds and storage ___10___ change rapidly.

1. A. As	B. It	C. As it	D. That
2. A. what	B. which	C. where	D. when
3. A. performs	B. carries	C. makes	D. integrates
4. A. decimal	B. binary	C. Arabian	D. American
5. A. technique	B. technology	C. techniques	D. technologies
6. A. instead of	B. rather than	C. but also	D. as well
7. A. Application	B. System	C. Word	D. Excel
8. A. put	B. made	C. conducted	D. classified
9. A. precious	B. progress	C. proceeding	D. precise
10. A. capacities	B. capable	C. capabilities	D. capacity

Translation

Ⅷ. Translate the following into Chinese.

1. By using various coding techniques, groups of bits can be made to represent not only binary numbers but also other discrete symbols.

2. System software includes not only the complex programs used by technicians to create application software in the first place but also the organizational programs needed to start up the computer and govern its use of other programs.

3. Data are numbers and other binary-code information that are operated on to achieve required computational results.

4. Rather than arithmetically or logically manipulating characters, a computer may concatenate strings of characters, replace some characters with others, or otherwise manipulate character strings.

5. Software applications like word processing, electronic spreadsheets, database management programs, painting and drawing programs, desktop publishing, and so forth became commercially available, giving more people reasons to use a computer.

Chapter 2
Computer System Organization

Pre-reading Questions

1. What is a computer system?
2. How many levels of memory hierarchy do you know? Name at least three of them.
3. Identify the types of data transfer modes.

In this chapter, we will examine the organization of basic computer systems, which consist of computer organization introduction, system buses, CPU organization, memory subsystem organization and I/O subsystem organization.

2.1 Computer Organization Introduction

A computer system consists of hardware system and software system. The hardware of the computer is usually divided into three major parts or three primary subsystems: the CPU, the memory subsystem, and the I/O subsystem. Figure 2-1 shows the generic organization of these components.

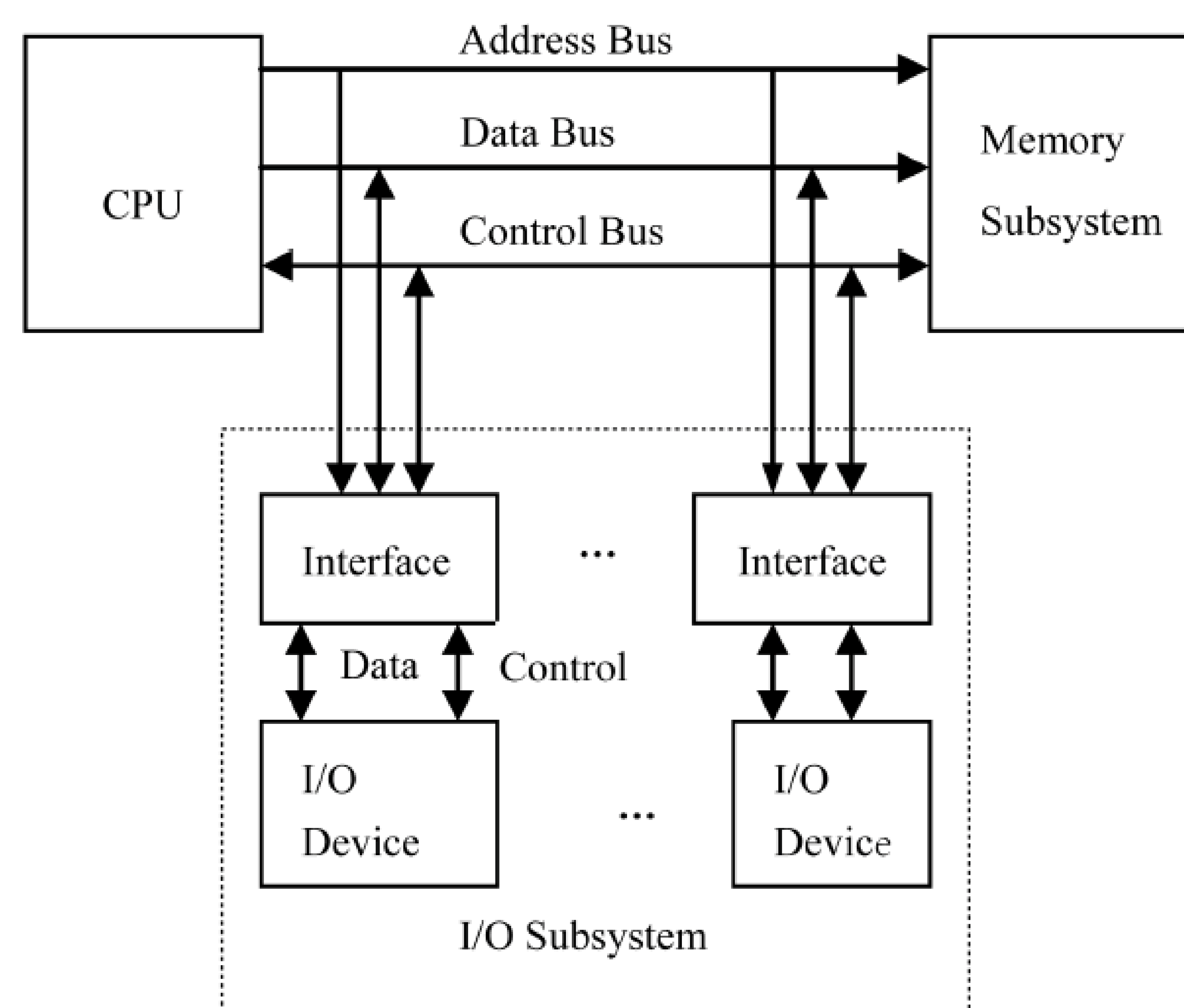


Figure 2-1 Computer Organization

The CPU performs many operations and controls computer. The memory subsystem is used to store the program being executed by the CPU, along with the program's data. The I/O subsystem

allows the CPU to interact with input and output devices such as the keyboard and monitor of a personal computer.

The components of a computer are connected to the buses. So we first describe the system buses.

2.2 System Buses

Physically, a bus is a set of wires. They are used to connect the components in the computer system. To send information from one component to another, the source component outputs data onto the bus. The destination component then inputs this data from the bus. As the complexity of a computer system increases, it becomes more efficient at using buses rather than direct connections between every pair of devices.

2.2.1 Address Bus

The system buses shown in Figure 2-1 have three buses. The uppermost bus in the figure is the address bus. When the CPU reads data or instructions from or writes data to memory, it must specify the address of the memory location it wishes to access. It outputs this address to the address bus, memory inputs this address from the address bus and uses it to access the proper memory location. When accessing an I/O device, the CPU places the address of the device on the address bus. Unlike the other buses, the address bus always receives data from the CPU, and the CPU never reads the address bus.

2.2.2 Data Bus

Data is transferred via the data bus. When the CPU fetches data from memory, it first outputs the memory address on its address bus. Then memory outputs the data onto the data bus, the CPU can then read the data from the data bus. When writing data to memory, the CPU first outputs the address onto the address bus, and then outputs the data onto the data bus. Memory then reads and stores the data at the proper location. The processes for reading data from and writing data to the I/O devices are similar.

2.2.3 Control Bus

The control bus is different from the other two buses. The address bus consists of *n* lines, which combine to transmit one *n*-bit address value. Similarly, the lines of the data bus work together to transmit a single, multibit value. In contrast, the control bus is a collection of individual control signals. These signals indicate whether data is to be read into or written out of the CPU, whether the CPU is accessing memory or an I/O device, and whether the I/O device or memory is ready to transfer data. Most of these signals are output from the CPU to the memory and I/O subsystems, and a few are output by these subsystems to the CPU.

2.3 CPU Organization

The part of the computer that performs the bulk of data processing operations is called the central processing unit and is referred to as the CPU. In microcomputer, it is often called the microprocessor. The CPU is made up of three major parts, as shown in Figure 2-2.

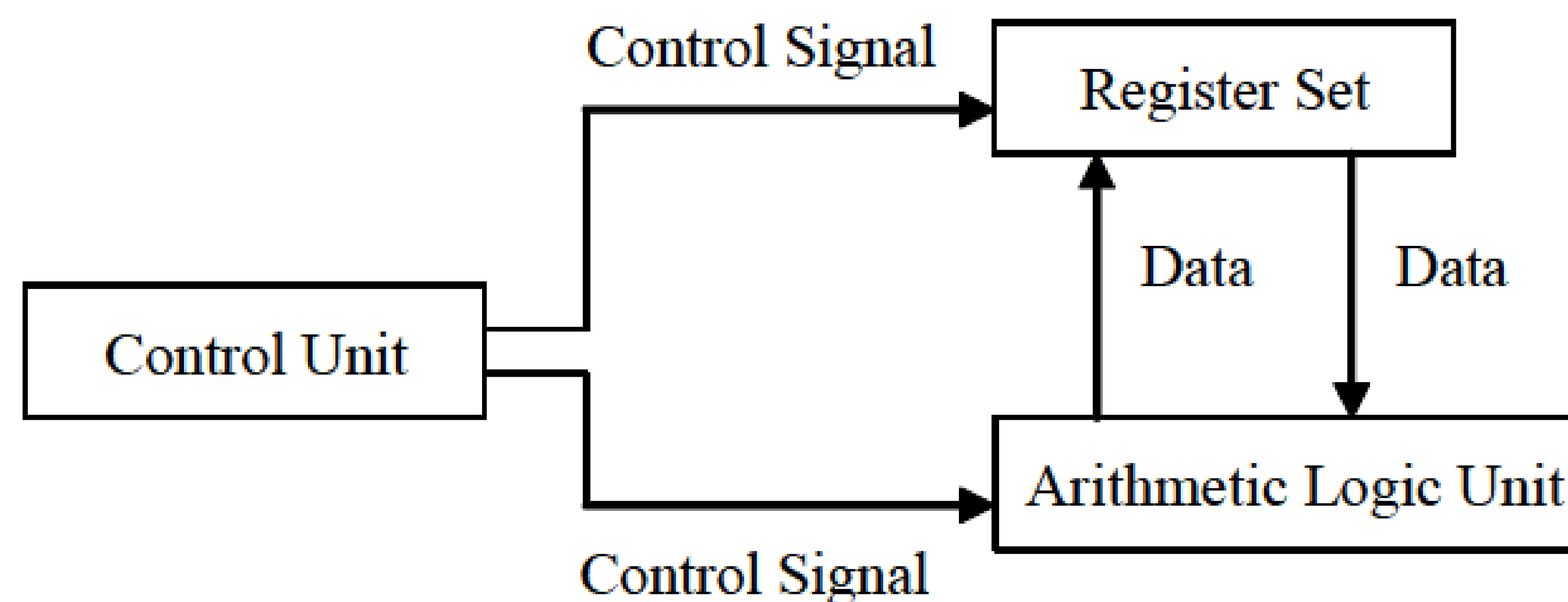


Figure 2-2 CPU Internal Organization

2.3.1 Register Set

The register set stores intermediate data used during the execution of the instructions. The register set, as its name implies, includes a set of registers and a bus or other communication mechanism. The register set also contains other registers that are not directly accessible by the programmer. The relatively simple CPU includes registers to latch the address being accessed in memory and a temporary storage register, as well as other registers that are not a part of its instruction set architecture.

2.3.2 Arithmetic Logic Unit

The arithmetic logic unit, or ALU, performs most arithmetic and logical operations, such as adding values. It receives its operands from the register set of the CPU and stores its results back in the register set. Since the ALU must complete its operations within a single clock cycle, it is constructed by using only combinatorial logic.

2.3.3 Control Unit

The control unit, or CU, supervises the transfer of information among the registers and instructs the ALU as to which operation to perform. This unit generates the internal control signals that cause registers to load data, increment or clear their contents, and output their contents, as well as cause the ALU to perform the correct function. The control unit receives some data values from the register set, which it uses to generate the control signals. This data includes the instruction code and the values of some flag registers. The control unit also generates the signals for the system control bus, such as the READ, WRITE, and IO/M# signals. A microprocessor typically performs a sequence of operations to fetch, decode, and execute an instruction. By asserting these internal and external control signals in the proper sequence, the control unit causes the CPU and

the rest of the computer to perform the operation needed to correctly process instructions.

The CPU performs a variety of functions dictated by the type of instructions that are incorporated in the computer. Computer architecture is sometimes defined as the computer structure and behavior as seen by the programmer that uses machine language instructions. This includes the instruction format, addressing modes, the instruction set, and the general organization of the CPU register.

2.4 Memory Subsystem Organization

Memory is also known as internal memory or main memory. It refers to the circuits in the computer that hold whatever programs and data available for immediate use by the CPU. In this section we will review the different types of memory and discuss the memory hierarchy.

2.4.1 Types of Memory

There are two major types of memory: Random Access Memory (RAM) and Read Only Memory (ROM).

1. RAM

RAM, also called read/write memory, can be used to store data that changes. Unlike ROM, RAM is erasable, temporary, and volatile, meaning that it blanks out each time you turn the computer off. RAM is essential to a computer because it provides rapidly accessible circuitry where the CPU can run your programs and process your data. You may have dozens of computer games or other application software programs stored on a disk, but the computer cannot use them there; when you select a program to run, the computer must first copy it from the disk to RAM where the CPU can access it. It’s particularly important to remember that all the data you type into the computer goes directly to RAM. If you want to keep your data permanently, you must save it by copying it to a secondary storage device such as a disk. The amount of RAM in a computer can be very important. If your computer has too little memory, complicated programs will run slowly or won’t run at all. To stretch the amount of RAM, an operating system can use a strategy called virtual memory that reserves part of the hard disk for use as an extension to RAM. In a computer with virtual memory, less-used parts of programs are shifted from RAM to a hard disk and are moved back only when needed. Although virtual memory may allow your computer to run large programs, it will reduce the amount of secondary storage and may cause programs to run slowly because it takes time to swap data to and from the hard disk.

There are two types of RAM chips, which are differentiated by how they maintain their data.

(1) DRAM

Dynamic RAM or DRAM chips are like leaky capacitors. Initially data is stored in the DRAM chip, charging its memory cells to their maximum values. The charge slowly leaks out and would

eventually go too low to represent valid data. Before this happens, however, refresh circuitry reads the contents of DRAM and rewrites the data to its original locations, thus restoring the memory cells to their maximum charges. DRAM is used to construct RAM in personal computers.

(2) SRAM

Static RAM, or SRAM, is more like a register than a leaky capacitor. Once data is written to SRAM, its contents stay valid, and it does not have to be refreshed. SRAM is faster than DRAM, but it is also much more expensive. The cache memory in personal computers is constructed from SRAM.

2. ROM

ROM chips are designed for applications in which data is only read. These chips are programmed with data by an external programming unit before they are added to the computer system. Once this is done, the data usually does not change. A ROM chip is well suited for this purpose because it is nonvolatile, meaning that the instructions recorded in it do not disappear when the power is off. When you turn the computer on, the permanent software in ROM boots the computer. To boot the computer, ROM first tells the CPU to determine what input, output, and storage devices happen to be attached to your computer. Then it instructs the CPU to check a disk to see if it contains operating system software that will let you start giving commands. The ROM in a typical personal computer also has a variety of self-test routines to make it easier to diagnose and repair hardware failures.

There are several types of ROM chips, which are differentiated by how and how often they can be programmed.

(1) Masked ROM

A masked ROM, or simply a ROM, is programmed with data as the chip is fabricated. The mask used to create the chip is designed with the required data hard-wired into it. These chips are useful for consumer appliances, where large numbers of units are produced, and, once installed, data will not be changed.

(2) PROM

A PROM is programmable ROM. Unlike the ROM, the PROM has a series of internal connections similar to fuses. Programming data into the PROM essentially blows the proper fuses so that each word of memory stores the correct value. Because these fuses cannot be restored once they are blown, PROMs can only be programmed once.

(3) EPROM

An EPROM is an erasable PROM. As its name implies, an EPROM can be programmed like a PROM, but its contents can be erased and the chip reprogrammed. These chips have a small clear window on their faces. The chip is erased by being placed under ultraviolet light that causes the capacitors to leak their charge, thus we can reset the chip’s contents. When in use, the window is usually covered with opaque tape to prevent any ultraviolet rays in room light or sunlight from inadvertently destroying the contents of the chip. Because they are erasable, but are used in

applications where their data does not change, EPROMs are typically used in product development labs and in prototypes.

(4) EEPROM or E^2PROM

An EEPROM, sometimes denoted as E^2PROM, is an electrically erasable PROM. It works like an EPROM, except that its contents are erased and reprogrammed electrically, rather than by using ultraviolet light. Unlike the EPROM, which must be entirely erased and then reprogrammed, it is possible to modify individual locations of the EEPROM while leaving other locations unchanged. Also, it takes only seconds to reprogram an EEPROM, but it takes about 20 minutes to erase an EPROM. One common use for EEPROMs is the basic input/output system, or BIOS, of personal computers.

(5) Flash EEPROM

A special type of EEPROM called a flash EEPROM is electrically erasable in blocks of data, rather than individual locations. It is well suited for applications that write blocks of data and can be used as a solid stated hard disk. It is also used for data storage in digital cameras. Flash EEPROM can only be rewritten with data a finite number of times, which currently limits its widespread use in computer systems.

2.4.2 Memory Hierarchy

The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high-capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory accessible to the high-speed processing logic. The components in a typical memory hierarchy are shown in Figure 2-3.

The memory unit that communicates directly with the CPU is called the main memory. Devices that provide backup storage are called auxiliary memory. The most common auxiliary memory devices used in computer systems are magnetic disks and tapes. They are used for storing system programs, large data files, and other backup information. Only programs and data currently needed by the CPU reside in main memory. All other information is stored in auxiliary memory and transferred to main memory when needed.

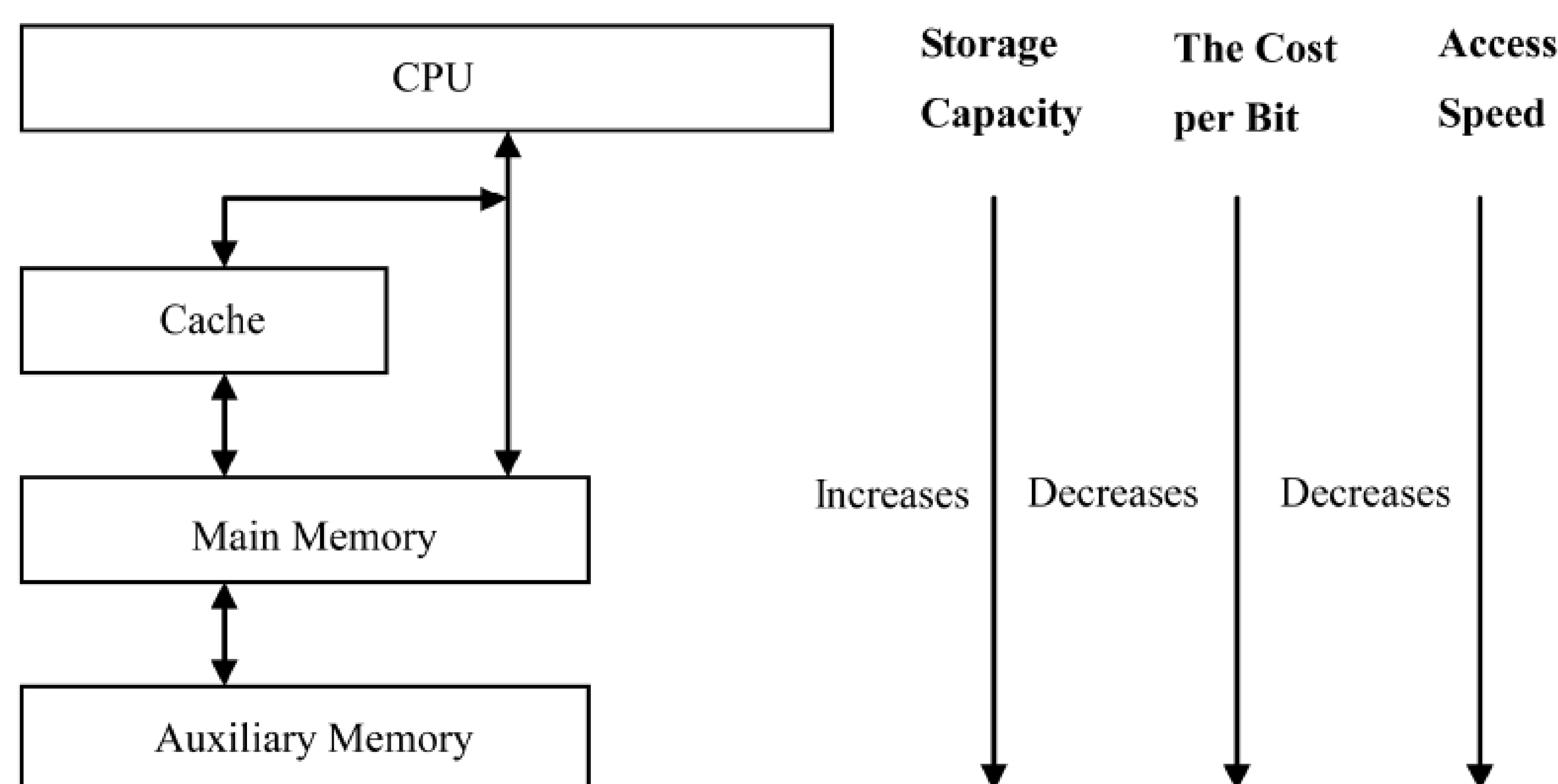


Figure 2-3　A Typical Memory Hierarchy

A special very-high-speed memory called a cache is sometimes used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate.The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor logic. CPU logic is usually faster than main memory access time, with the result that processing speed is limited primarily by the speed of main memory. A technique used to compensate for the mismatch in operating speeds is to employ an extremely fast, small cache between the CPU and main memory whose access time is close to processor logic clock cycle time. The cache is used for storing segments of programs currently being executed in the CPU and temporary data frequently needed in the present calculations. By making programs and data available at a rapid rate, it is possible to increase the performance rate of the computer.

Figure 2-3 shows the CPU has direct access to both cache and main memory but not to auxiliary memory. In a memory hierarchy system, each component is involved with a different level. The Auxiliary memory which has a large storage capacity, is relatively inexpensive, but has low access speed compared with main memory. The cache memory is very small, relatively expensive, and has very high access speed. Thus as the memory access speed increases, so does its relative cost. The overall goal of using a memory hierarchy is to obtain the highest-possible average access speed while minimizing the total cost of the entire memory system.

2.5 I/O Subsystem Organization

In this section, we will describe the I/O devices, I/O interface, and modes of transfer.

2.5.1 I/O Devices

1. Keyboards

A computer's keys are generally divided into four clusters: alphanumeric keys, function keys, cursor keys, and the numeric keypad. Alphanumeric keys include letters, numbers, and punctuation marks. They are arranged much like the keys on a typewriter. Function keys are labeled F1, F2, and so on up to F12 or F15. They can be used for giving common commands such as "Print"or "Quit program". The precise purpose of any function key varies from one program to another. Cursor keys are used to move the cursor around on the screen. The cursor is the little blinking symbol that indicates where things will happen next on the screen. When you are typing, the cursor always blinks just to the right of the last character you typed. Cursor keys include the arrow keys for moving up, down, left, and right, as well as the PgDn, PgUp, Home, and End keys. The numeric keypad includes the mathematical keys found on a standard calculator.

2. Mouse and Touch Screens

Mice are popular because it is easier to point than to type, and because the arrow keys don't work very well for drawing pictures or moving things on the screen. A mouse consists of a ball

mounted under a plastic housing with one or more buttons on top. As you move the mouse around the tabletop, sensors inside register the rolling of the ball and move the cursor around the screen to match. There are three basic ways of giving commands with the mouse. First, you can click the button to identify something — perhaps to indicate which part of a drawing you want to change. Second, you can drag the mouse; that is, you can hold the button down while you move the mouse. Dragging can be used to move a drawing across the screen. The third way to give a command is to double-click the mouse's button by pressing it twice within about a half second. Double-clicking is used to select things on the screen.

The simplest pointing tool of all, of course, is the finger. In fact, touch screens are widely used in department-store advertising displays, information kiosks, lottery game machines, and other places where users are not expected to have much familiarity with computers. On these machines you simply touch the part of the display screen you want to select, just as you might push a vending machine's button. Depending on the sensing method used by the touch screen, your finger might interrupt a network of infrared rays protected across the screen's surface. This would tell the computer where you pointed.

3. Monitor

A monitor, or display screen, provides a convenient but temporary way to view information. The earliest computer monitors were simply converted television sets. While ordinary TVs are still used for some video games, most computer programs today demand higher-quality monitors. The quest for better monitors has taken two paths: the improvement of TV-like screens and the development of flat screens.

4. Printers

Different kinds of computer printers use surprisingly different technologies. Some printers squirt ink, some apply heat to sensitive paper, others hammer inked ribbons, and still others create images with lasers. Laser printers work by reflecting a laser beam from a rapidly rotating octagonal mirror onto a light-sensitive roller. Ink-jet printers work by squirting tiny droplets of liquid ink at the paper. Dot-matrix printer features a movable print head containing a row of tiny pins. The pins push an inked ribbon against the paper, producing a matrix (or pattern) of dots. As the print head moves back and forth across the page, the dots can form either letters or graphics.

5. Modem

A modem, short for modulator-demodulator, is used for communicating between computers. The modem converts a computer's electronic impulses to a form that can be transmitted over a telephone line. When the signal reaches the destination computer, another modem reconverts the signal to computer understandable form.

2.5.2 I/O Interface

Input/output interface provides a method for transferring information between internal storage

and external I/O devices. Peripherals connected to a computer need special communication links for interfacing them with the central processing unit. The purpose of the communication link is to resolve the differences that exist between the central computer and each peripheral. The major differences are in the following:

- Peripherals are electromechanical and electromagnetic devices and their manner of operation is different from the operation of the CPU and memory, which are electronic devices. Therefore, a conversion of signal values may be required.
- The data transfer rate of peripherals is usually slower than the transfer rate of the CPU, and consequently, a synchronization mechanism may be needed.
- Data codes and formats in peripherals differ from the word format in the CPU and memory.
- The operating modes of peripherals are different from each other and each must be controlled so as not to disturb the operation of other peripherals connected to the CPU.

To resolve these differences, computer systems include special hardware components between the CPU and peripherals to supervise and synchronize all input and output transfers. These components are called interface units because they interface between the processor bus and the peripheral device. I/O interface shows in Figure 2-1. Each peripheral device has associated with an interface unit. Each interface decodes the address and control received from the I/O bus, interprets them for the peripheral, and provides signals for the peripheral controller. It also synchronizes the data flow and supervises the transfer between peripheral and processor.

2.5.3 Modes of Transfer

Binary information received from an external device is usually stored in memory for later processing. Information transferred from the central computer into an external device originates in the memory unit. The CPU merely executes the I/O instructions and may accept the data temporarily, but the ultimate source or destination is the memory unit. Data transfer between the central computer and I/O devices may be handled in a variety of modes. Some modes use the CPU as an intermediate path; the others transfer the data directly to and from the memory unit. Data transfer to and from peripherals may be handled in one of three possible modes: programmed I/O, interrupts, and direct memory access (DMA).

1. Programmed I/O

Programmed I/O operations are the result of I/O instructions written in the computer program. Usually, the transfer is to and from a CPU register and peripheral. Other instructions are needed to transfer the data to and from CPU and memory. Transferring data under program control requires constant monitoring of the peripheral by the CPU. Once a data transfer is initiated, the CPU is required to monitor the interface to see when a transfer can again be made. It is up to the programmed instructions executed in the CPU to keep close tabs on everything that is taking place in the interface unit and the I/O device.

In the programmed I/O method, the CPU stays in a program loop until the I/O unit indicates that it is ready for data transfer. This is a time-consuming process since it keeps the processor busy needlessly. It can be avoided by using an interrupt facility and special commands to inform the interface to issue an interrupt request signal when the data is available from the device. In the meantime the CPU can proceed to execute another program. The interface meanwhile keeps monitoring the device. When the interface determines that the device is ready for data transfer, it generates an interrupt request to the computer. Upon detecting the external interrupt signal, the CPU momentarily stops the task it is processing, branches to a service program to process the I/O transfer, and then returns to the task it was originally performing.

2. Interrupts

An alternative to the CPU constantly monitoring the flag is to let the interface inform the computer when it is ready to transfer data. This mode of transfer uses the interrupt facility. While the CPU is running a program, it does not check the flag. However, when the flag is set, the computer is momentarily interrupted from proceeding with the current program and is informed of the fact that the flag has been set. The CPU deviates from what it is doing to take care of the input or output transfer. After the transfer is completed, the computer returns to the previous program to continue what it was doing before the interrupt.

The CPU responds to the interrupt signal by storing the return address from the program counter into a memory stack and then controls branches to a service routine that processes the required I/O transfer. The way that the processor chooses the branch address of the service routine varies from one unit to another. In principle, there are two methods for accomplishing this. One is called vectored interrupt and the other, non-vectored interrupt. In a non-vectored interrupt, the branch address is assigned to a fixed location in memory. In a vectored interrupt, the source that interrupts supplies the branch information to the computer. This information is called the interrupt vector. In some computers the interrupt vector is an address that points to a location in memory where the beginning address of the I/O service routine is stored.

3. Direct Memory Access (DMA)

Another aspect of computer system performance that can be improved is the transfer of data between memory and I/O devices. This is a common operation in computer systems. Loading programs or data files from disk into memory, saving files on disk, and accessing virtual memory pages on any secondary storage medium all fall into this category of operations.

Consider a typical system consisting of a CPU, memory, and one or more input/output devices. Assume one of the I/O devices is a disk drive and that the computer must load a program from this drive into memory. The CPU would read the first byte of the program and then write that byte to memory. Then it would do the same for the second byte and each succeeding byte, until it had loaded the entire program into memory.

This is, at best, inefficient. Loading data into the memory, and then writing data out of it, the

CPU significantly slows down the transfer. The CPU does not modify the data at all, so it only serves as an additional stop for data on the way to its final destination. The process would be much quicker if we could bypass the CPU and transfer data directly from the I/O device to memory. DMA exactly does that.

A DMA controller implements direct memory access in a computer system. It connects directly to the I/O device at one end and to the system buses at the other end. It also interacts with the CPU, both via the system buses and two new direct connections.

To transfer data from an I/O device to memory, the DMA controller first sends a bus request to the CPU by setting BR (Bus Request) to 1. When it is ready to grant this request, the CPU sets its bus grant signal, BG (Bus Grant), to 1. The CPU also tri-states its address, data, and control lines, thus truly granting control of the system buses to the DMA controller. The CPU will continue to tri-state its outputs as long as BR is asserted.

Now that the DMA controller has control of the system buses, it can perform the desired data transfers. To load data from an I/O device into memory, it asserts the appropriate I/O control signals and loads data from the I/O device into its internal DMA data register. Next, it writes this data to memory. To do this, it outputs the memory address onto the system's address bus and the data onto the data bus. The DMA controller also asserts the appropriate signals on the system's control bus to cause memory to read the data. DMA controller then writes the second data value to the following memory location, continuing until it has transferred the entire block of data.

To understand how the DMA controller performs this transfer, we must examine its internal architecture. The DMA controller includes several registers. The DMA address register contains the memory address to be used in the data transfer. The DMA count register, sometimes called the word count register, contains the number of bytes of data to be transferred. The DMA control register accepts commands from the CPU. Most DMA controllers also have a status register. This register supplies information to the CPU. DMA controllers also usually include circuitry to abort the transfer if the I/O device is not ready in some predetermined amount of time. This is called timeout.

Technical Notes to the Text

1. bus，总线。计算机系统各部件间的一种电连接，信号及电源就是通过它传送的。信息可从多个源部件中的任何一个经总线传送到多个目标部件中的任意一个。总线由若干平行导线组成，分别传送地址、数据、同步信号、控制信息及电源等。总线的类型有如下几种。

- address bus，地址总线。一种单向总线，其上传输用来标识特定的存储单元或特定的输入/输出设备的数字信息。
- data bus，数据总线。在处理器、存储器及外部设备之间进行通信的信息通路。
- control bus，控制总线。一种运载用来调整系统运行的信号的总线。

2. flag register，标志寄存器。一种特殊用途的寄存器，该寄存器的各位按照执行指令期

间可能发生的规定条件来设置。

3. cache memory，高速缓冲存储器。速度比随机存储器高得多，它用于加快数据从随机存储器到中央处理器的流动速度。高速缓冲存储器利用一种算法预先估计中央处理器要从随机存储器中获取什么信息，并把它提前从随机存储器取出存到高速缓冲存储器中。当中央处理器需要信息时，首先在高速缓冲存储器中去查找。如果估计正确，这些信息就可以迅速传到中央处理器，从而相应地提高了计算机的运行速度。高速缓冲存储器有数据和指令两种。

4. virtual memory，虚拟存储器。在具有层次结构存储器的计算机中，为用户提供一个比主存容量大得多的可随机访问的地址空间的技术。虚拟存储技术使辅助存储器和主存储器密切配合，对用户来说，好像计算机具有一个容量比实际主存大得多的主存可供使用。

5. prototype，原型，样机。在系统开发中，适用于系统设计、性能及生产潜力评估的一种系统模型。

6. BIOS (basic input output system)，基本输入输出系统，属于 CP/M 和 DOS 操作系统的重要组成部分，负责具体处理计算机的输入和输出数据。

7. interface，接口。接口是指计算机和其他实体之间(如打印机或操作者)相互作用或交流的点。

8. cursor，光标。光标是指显示器上明亮的、闪动的、可移动指示物，用于标明某一字母可以进入、修改或删除的位置。

9. touch screen，触摸屏。一种允许用户通过触摸屏上某一区域以与数据处理系统进行交互操作的显示装置。

10. modem (modulator-demodulator)，调制解调器。一种对信号进行调制与解调的功能部件。调制解调器的功能之一就是使数字数据经过模拟传输设施进行传送。

11. peripheral，外设。外部设备或辅助设备，如与计算机连接工作的打印机、调制解调器或存储系统。

12. synchronization，同步。调整两个信号相应的有效瞬间，以使它们之间满足所需相对关系的过程。

13. computer architecture，计算机体系结构。计算机体系结构是指计算机系统的物理或硬件结构、各组成部分的属性及这些部分的相互联系。它可以分为系统体系结构和实现体系结构两个方面。前者从软件开发人员的角度看计算机系统的功能行为和概念结构；后者从计算机系统的价格和性能特征出发，考虑系统的结构和实现。也有人将计算机体系结构专指为系统体系结构，而将实现结构称为计算机组织。

14. instruction format，指令格式。计算机指令的表示形式。一条指令通常包含操作码、操作数来源、下一条指令的地址等信息。

15. addressing modes，寻址方式。寻址方式指生成计算机指令中实际使用的有效地址的方法。广泛使用的寻址方式有直接寻址、寄存器寻址、间接寻址、基址寻址、变址寻址和相对寻址等。

16. instruction set，指令系统，指令集。其是指一台计算机中的所有指令的集合。

17. interrupt，中断。计算机在执行程序的过程中，当遇到急需处理的事件时，暂停当前正在运行的程序，转去执行有关服务程序，处理完后自动返回源程序，这个过程称为中断。

18. vectored interrupt，向量中断。存储器的地址码是一串布尔量的序列，是一个布尔向

量，所以地址码称为向量地址。当 CPU 响应中断时，由硬件直接产生一个或多个固定地址(即向量地址)，由向量地址指出中断服务程序的入口，这种方法称为向量中断。

19. DMA (direct memory access)，直接存储器存取，是一种完全由硬件执行 I/O 变换的工作方式。在这种方式中，DMA 控制器从 CPU 中完全接管对总线的控制，数据交换不经过 CPU，而直接在内存储器和 I/O 设备之间进行。

Word Bank to the Text

A. Useful new words

generic	*adj*. 类的，一般的，普通的
execute	*v*. 执行，实行
uppermost	*adj*. 至上的，最高的，最主要的
specify	*v*. 指定，详细说明，列入清单
transmit	*v*. 传输，转送
similarly	*adv*. 同样地，类似地
mechanism	*n*. 机械装置，机构，机制
accessible	*adj*. 易接近的，可到达的
architecture	*n*. 结构
operand	*n*. 操作数
combinatorial	*adj*. 组合的
supervise	*v*. 监督，管理
generate	*v*. 产生，发生
sequence	*n*. 次序，顺序
dictate	*v*. 指令，指示，命令
hierarchy	*n*. 层次
random	*adj*. 任意的，随便的，随机的
volatile	*adj*. 飞行的，挥发性的，可变的
secondary	*adj*. 次要的，二级的
differentiate	*v*. 区别，区分
dynamic	*adj*. 动力的，动力学的，动态的
attach	*v*. 把……固定，把……附(在……上)
fabricate	*v*. 制作，构成
fuse	*n*. 保险丝，熔丝
erasable	*adj*. 可消除的，可抹去的
ultraviolet	*adj*. 紫外线的，紫外的
prototype	*n*. 原型
auxiliary	*adj*. 辅助的
alphanumeric	*adj*. 字母与数字混合编排的

kiosk	*n*. 书报亭，报摊
infrared	*adj*. 红外线的
squirt	*v*. 喷出
octagonal	*adj*. 八边形的，八角形的
synchronization	*n*. 同时(性)，[物]同步
initiate	*v*. 开始，发动
vector	*n*. [数]向量，矢量
implement	*v*. 贯彻，实现
predetermine	*v*. 预定，预先确定

B. Useful expressions

divide into	分成
along with	连同……一起
interact with	与……相合
receive from	收到
be different from	异于……
in contrast	相反，大不相同
be ready to	预备，即将
refer to as	认为与……有关
as well as	也，又
in the proper sequence	以适当的次序
be known as	被认为是
blank out	取消，作废
shift from	转移，移转
leak out	漏出
be suited for	适合
be attached to	连在……上，附属于
a variety of	一系列
be similar to	与……相似
compensate for	赔偿
have much familiarity with	与……熟悉
reflect onto	反射到
keep busy	使忙碌
mode of transfer	转移模式
deviate from	背离，偏离
in principle	原则上，大体上
fall into	分成，属于
now that	既然

C. Technical terms and proper names

memory subsystem	存储子系统
I/O subsystem	输入/输出子系统
bus	总线
system bus	系统总线
chip	芯片
address bus	地址总线
instruction	指令
memory location	存储单元
data bus	数据总线
control bus	控制总线
local bus	局部总线
microprocessor	微处理器
register set	寄存器组
arithmetic logic unit (ALU)	运算器
clock cycle	时钟周期
control unit	控制器
computer architecture	计算机体系结构
instruction format	指令格式
addressing mode	寻址方式
instruction set	指令集
internal memory	内存储器
main memory	主存储器
random access memory (RAM)	随机存取存储器
read only memory (ROM)	只读存储器
secondary storage=auxiliary memory	辅助存储器
virtual memory	虚拟存储器
dynamic RAM (DRAM)	动态 RAM
refresh circuitry	刷新电路
static RAM (SRAM)	静态 RAM
cache memory	高速缓冲存储器
masked ROM	掩膜 ROM
PROM	可编程 ROM
EPROM	可擦写 PROM
ultraviolet light	紫外线
EEPROM / E^2PROM	电擦写 PROM
basic input/output system (BIOS)	基本输入/输出系统
flash EEPROM	快闪存储器

memory hierarchy	存储器体系结构
storage capacity	存储容量
keyboard	键盘
alphanumeric key	字母数字键
function key	功能键
cursor key	光标键
numeric keypad	数字键
mouse	鼠标
touch screen	触摸屏
infrared ray	红外线
monitor	监视器
display screen	显示屏
laser printer	激光打印机
ink-jet printer	喷墨打印机
dot-matrix printer	点阵式打印机
modem	调制解调器
input-output interface (I/O interface)	输入输出接口(I/O 接口)
peripheral	外部(外围)设备，外设
interrupt	中断
program counter	程序计数器
vectored interrupt	向量中断
non-vectored interrupt	非向量中断
interrupt vector	中断向量
direct memory access (DMA)	直接存储器存取
timeout	超时

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. The computer hardware consists of three major parts which are the CPU, the memory subsystem, and ________.

2. There are two major types of memory: Random Access Memory (RAM) and ________.

3. The cache memory in personal computers is constructed from ________.

4. ________ provides a method for transferring information between internal storage and external I/O devices.

5. Data transfer between the central computer and I/O devices may be handled in a variety of

modes. These modes are programmed I/O, ________, and direct memory access (DMA).

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The data bus always receives data from the CPU, and the CPU never reads the data bus. (   )

2. Main memory holds whatever programs and data are available for immediate use by the CPU. (   )

3. Dynamic RAM does not have to be refreshed. (   )

4. Dot-matrix printer works by squirting tiny droplets of liquid ink at the paper. (   )

5. The auxiliary memory is very small, relatively expensive, and has very high access speed. (   )

Ⅲ. Match each of the following terms with the appropriate definition.

ALU	RAM	E^2PROM	DMA

1. ________ An electrically erasable PROM.
2. ________ A transfer mode that can be improved the transfer of data between memory and I/O devices.
3. ________ The unit which performs most arithmetic and logical operations.
4. ________ Memory that is erased when the computer is turned off.

Ⅳ.Translate the following into Chinese.

1. system bus 2. virtual memory 3. computer architecture
4. instruction set 5. direct memory access

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

auxiliary	generate	implement
specify	initiate	accessible
differentiate	volatile	execute

1. The Employment Minister said the reforms would ______ new jobs.
2. The international oil markets have been highly ________ since the early 1970s.
3. They wanted to ______ a discussion on economics.
4. The government promised to ________ a new system to control financial loan institutions.
5. One group claimed to have ________ the American hostage.
6. He is proud that his wife is ________ to reason.
7. The government's first concern was to augment the army and ________ forces.
8. A child may not ________ between his imagination and the real world.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can “generate” and fill in the sentences with the right word.

Example: This unit **generates** the internal control **signals** that cause registers to load data.

1. With its ability to generate numerous se________ quickly, this new type of weed soon became the disaster of the land.

2. They’ve just built a new nuclear plant to generate more el________ to meet the demands of the local people.

3. Mid-sized cities like Bangalore are now the Silicon Valleys of India — their workers generate de________ for the very products that they produce.

4. The government will put forward a new program which will strengthen local economies, generate jo________ and improve the quality of people’s lives.

5. Mike’s absence from the class that morning didn’t generate any at________.

6. The vast amount of pu______ they generated was immediately reflected in sales.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

A computer system __1__ of hardware system and software system. The hardware of the computer is usually divided into three major parts or three __2__ subsystems: the CPU, the memory subsystem, and the I/O __3__. The CPU is made up of three major parts, Register Set, the __4__ logic unit, or ALU, and Control Unit. It performs many operations and controls computer. Memory is also known as __5__ memory or main memory, which is cataloged into two major types of memory: Random Access Memory (RAM) and Read Only Memory (ROM). It refers to the __6__ in the computer that hold whatever programs and data are available __7__ immediate use by the CPU, along with the program’s data. Computer systems include special hardware __8__ between the CPU and peripherals to supervise and synchronize all input and output transfers. These components are called __9__ units because they interface between the processor bus and the peripheral device. The I/O subsystem allows the CPU to __10__ with input and output devices.

1. A. consists	B. makes up	C. constitutes	D. comprise
2. A. premier	B. primary	C. preliminary	D. elementary
3. A. system	B. machine	C. subsystem	D. device
4. A. mathematic	B. authoritive	C. arithmetic	D. authoritative
5. A. external	B. exterior	C. interior	D. internal
6. A. circuits	B. wires	C. lines	D. hardware
7. A. by	B. for	C. with	D. in

8. A. software	B. setting	C. listing	D. components
9. A. singular	B. dual	C. interface	D. compact
10. A. handle	B. interact	C. respond	D. link

Translation

Ⅷ. Translate the following into Chinese.

1. By asserting these internal and external control signals in the proper sequence, the control unit causes the CPU and the rest of the computer to perform the operation needed to correctly process instructions.

2. In a computer with virtual memory, less-used parts of programs are shifted from RAM to a hard disk and are moved back only when needed.

3. A technique used to compensate for the mismatch in operating speeds is to employ an extremely fast, small cache between the CPU and main memory whose access time is close to processor logic clock cycle time.

4. The data transfer rate of peripherals is usually slower than the transfer rate of the CPU, and consequently, a synchronization mechanism may be needed.

5. In some computers the interrupt vector is an address that points to a location in memory where the beginning address of the I/O service routine is stored.

Chapter 3
Computer System Architecture

Pre-reading Questions

1. How many types of parallel processing do you know? Name at least four of them.
2. What is a pipelining?
3. What is a RISC?

In this chapter, we will introduce parallel processing, pipelining, vector processing and RISC.

3.1 Parallel Processing

Parallel processing is a term used to denote a large class of techniques that are used to provide simultaneous data-processing tasks for the purpose of increasing the computational speed of a computer system. Instead of processing each instruction sequentially as in a conventional computer, a parallel processing system is able to perform concurrent data processing to achieve faster execution time. For example, while an instruction is being executed in the ALU, the next instruction can be read from memory. The system may have two or more ALUs and be able to execute two or more instructions at the same time. Furthermore, the system may have two or more processors operating concurrently. The purpose of parallel processing is to speed up the computer processing capability and increase its throughput, that is, the amount of processing that can be accomplished during a given interval of time. The amount of hardware increases with parallel processing, and with it, the cost of the system increases. However, technological development has reduced hardware costs to the point where parallel processing techniques are economically feasible.

Parallel processing can be viewed from various levels of complexity. At the lowest level, we distinguish between parallel and serial operations by the type of registers used. Shift registers operate in serial fashion one bit at a time, while registers with parallel load operate with all the bits of the word simultaneously. Parallel processing at a higher level of complexity can be achieved by having a multiplicity of functional units that perform identical or different operations simultaneously. Parallel processing is established by distributing the data among the multiple functional units. For example, the arithmetic logic and shift operations can be separated into three units and the operands diverted to each unit under the supervision of a control unit.

There are a variety of ways that parallel processing can be classified. It can be considered from the internal organization of the processors, from the interconnection structure between processors, or from the flow of information through the system. One classification introduced by

M. J. Flynn considers the organization of a computer system by the number of instructions and data items that are manipulated simultaneously. The normal operation of a computer is to fetch instructions from memory and execute them in the processor. The sequence of instructions read from memory constitutes an instruction stream. The operations performed on the data in the processor constitute a data stream. Parallel processing may occur in the instruction stream, in the data stream, or in both. Flynn's classification divides computers into four major groups as follows:

- Single instruction, single data stream (SISD);
- Single instruction, multiple data stream (SIMD);
- Multiple instruction, single data stream (MISD);
- Multiple instruction, multiple data stream (MIMD).

SISD represents the organization of a single computer containing a control unit, a process unit, and a memory unit. Instructions are executed sequentially and the system may or may not have internal parallel processing capabilities. Parallel processing in this case may be achieved by means of multiple functional units or by pipeline processing.

SIMD represents an organization that includes many processing units under the supervision of a common control unit. All processors receive the same instruction from the control unit but operate on different items of data. The shared memory unit must contain multiple modules so that it can communicate with all the processors simultaneously.

MISD structure is only of theoretical interest since no practical system has been constructed using this organization.

MIMD organization refers to a computer system capable of processing several programs at the same time. Most multiprocessor and multicomputer systems can be classified in this category.

Flynn's classification depends on the distinction between the performance of the control unit and the data-processing unit. It emphasizes the behavioral characteristics of the computer system rather than its operational and structural interconnections.

3.2 Pipelining

Pipelining is a technique of decomposing a sequential process into suboperations, with each subprocess being executed in a special dedicated segment that operates concurrently with all other segments. A pipeline can be visualized as a collection of processing segments through which binary information flows. Each segment performs partial processing dictated by the way the task is partitioned. The result obtained from the computation in each segment is transferred to the next segment in the pipeline. The final result is obtained after the data have passed through all segments. The name "pipeline" implies a flow of information analogous to an industrial assembly line. It is characteristic of pipelines that several computations can be in process in distinct segments at the same time. The overlapping of computation is made possible by associating a register with each segment in the pipeline. The registers provide isolation between each segment so that each can operate on distinct data simultaneously.

Perhaps the simplest way of viewing the pipeline structure is to imagine that each segment consists of an input register followed by a combinational circuit. The register holds the data and the combinational circuit performs the suboperation in the particular segment. The output of the combinational circuit in a given segment is applied to the input register of the next segment. A clock is applied to all registers after enough time has elapsed to perform all segment activities. In this way the information flows through the pipeline one step at a time.

The pipeline organization will be demonstrated by means of a simple example. Suppose that we want to perform the combined multiply and add operations with a stream of numbers.

$$A_i \times B_i + C_i \qquad i=1, 2, 3, 4$$

Each suboperation is to be implemented in a segment within a pipeline. Each segment has one or two registers and a combinational circuit as shown in Figure 3-1. R_1 through R_5 are registers that receive new data with every clock pulse. The multiplier and adder are combinational circuits. The suboperations performed in each segment of the pipeline are as follows:

$R_1 \leftarrow A_i$, $R_2 \leftarrow B_i$

$R_3 \leftarrow R_1 \times R_2$, $R_4 \leftarrow C_i$

$R_5 \leftarrow R_3 + R_4$

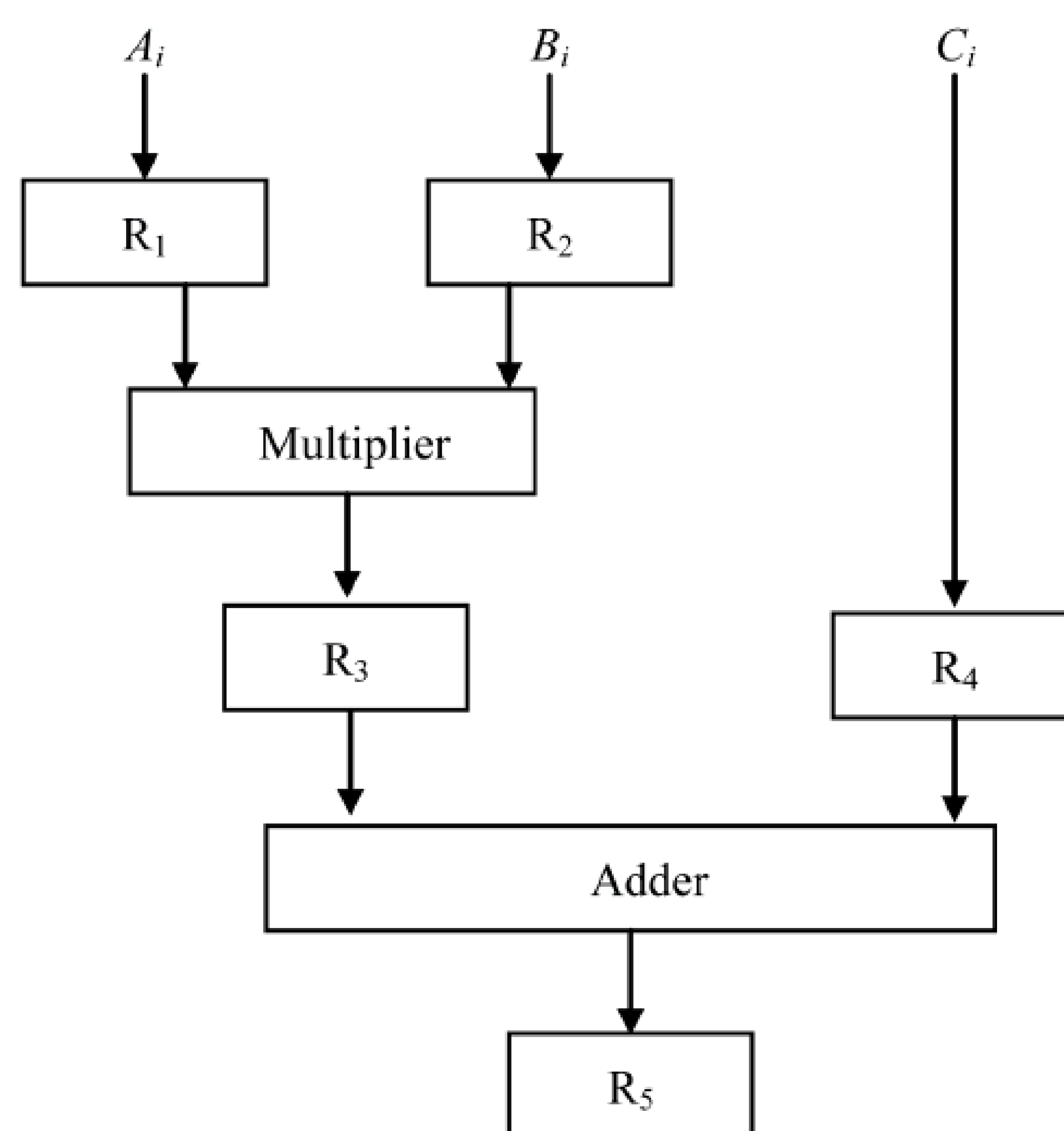


Figure 3-1　Pipeline Processing

The five registers are loaded with new data every clock pulse. The effect of each clock is shown in Table 3-1. The first clock pulse transfers A_1 and B_1 into R_1 and R_2.The second clock pulse transfers the product of R_1 and R_2 into R_3 and C_1 into R_4. The same clock pulse transfers A_2 and B_2 into R_1 and R_2. The third clock pulse operates on all three segments simultaneously. It places A_3 and B_3 into R_1 and R_2, transfers the product of R_1 and R_2 into R_3, transfers C_2 into R_4, and places the sum of R_3 and R_4 into R_5. It takes three clock pulses to fill up the pipe and receive the first output from R_5. From there on, each clock produces a new output and moves the data one step down the pipeline. This happens as long as new input data flow into the system. When no more input data are available, the clock must continue until the last output emerges out of the pipeline.

Table 3-1 Content of Register and Process Steps

Clock Pulse	Segment 1		Segment 2		Segment 3
	R_1	R_2	R_3	R_4	R_5
1	A_1	B_1			
2	A_2	B_2	$A_1 \times B_1$	C_1	
3	A_3	B_3	$A_2 \times B_2$	C_2	$A_1 \times B_1 + C_1$
4	A_4	B_4	$A_3 \times B_3$	C_3	$A_2 \times B_2 + C_2$
5			$A_4 \times B_4$	C_4	$A_3 \times B_3 + C_3$
6					$A_4 \times B_4 + C_4$

3.3 Vector Processing

There is a class of computational problems that are beyond the capabilities of a conventional computer. These problems are characterized by the fact that they require a vast number of computations that will take a conventional computer days or even weeks to complete. In many science and engineering applications, the problems can be formulated in terms of vectors and matrices that lend themselves to vector processing.

Computers with vector processing capabilities are in demand in specialized applications. The following are representative application areas where vector processing is of the utmost importance:

- Long-range weather forecasting;
- Petroleum explorations;
- Seismic data analysis;
- Medical diagnosis;
- Aerodynamics and space flight simulations;
- Artificial intelligence and expert system;
- Mapping the human genome;
- Image processing.

Without sophisticated computer, many of the required computations cannot be completed within a reasonable amount of time. To achieve the required level of high performance it is necessary to utilize the fastest and most reliable hardware and apply innovative procedures from vector and parallel processing techniques.

Many scientific problems require arithmetic operations on large arrays of numbers. These numbers are usually formulated as vector and matrices of floating-point numbers. A vector is an ordered set of a one-dimensional array of data items. A vector $\boldsymbol{X}$ of length n is represented as a row vector by $\boldsymbol{X}=[X_1, X_2, X_3 \ldots X_n]$. It may be represented as column vector if the data items are listed in a column. A conventional sequential computer is capable of processing operands once at a time. Consequently, operations on vectors must be broken down into single computations with subscripted

variables. The element X_i of vector ***X*** is written as X(I) and the index I refers to a memory address or register where the number is stored. To examine the difference between a conventional scalar processor and a vector processor, consider the following Fortran DO loop:

```
      DO ab I=1,100
  Ab   C(I) = B(I)+A(I)
```

This is a program for adding two vectors ***A*** and ***B*** of length 100 to produce vector ***C***. This is implemented in machine language by the following sequence of operations:

```
       Initialize I=0
Ab     Read    A(I)
       Read    B(I)
       Store C(I)=A(I)+B(I)
       Increment    I=I+1
       If I≤100 go to ab
       Continue
```

This constitutes a program loop that reads a pair of operands from arrays A and B and performs a floating-point addition. The loop control variable is then updated and the steps repeat 100 times.

A computer capable of vector processing eliminates the overhead associated with the time it takes to fetch and execute the instructions in the program loop. It allows operations to be specified with a single vector instruction of the form

C(1:100)=A(1:100)+B(1:100)

The vector instruction includes the initial address of the operands, the length of the vector, and the operation to be performed, all in one composite instruction.

A possible instruction format for a vector instruction is shown in Figure 3-2. This is essentially a three-address instruction with three fields specifying the base address of the operands and an additional field that gives the length of the data items in the vectors. This assumes that the vector operands reside in memory. It is also possible to design the processor with a large number of registers and store all operands in registers prior to the addition operation. In that case the base address and length in the vector instruction specify a group of CPU registers.

Operation code	Base address source 1	Base address source 2	Base address destination	Vector length

Figure 3-2　Instruction Format for Vector Processor

3.4　Reduced Instruction Set Computer (RISC)

The world of microprocessors and CPU can be divided into two parts: complex instruction set computers using CISC processors and reduced instruction set computers with RISC processors. Both seek to improve system performance, though paradoxically using directly opposite approaches.

The first microprocessors ever developed were very simple processors with very simple

instruction sets. As microprocessors became more complex, more instructions were incorporated into their instruction sets. Current CISC microprocessor instruction sets may include over 300 instructions. Some of these instructions, such as register moves, are used frequently; others are very specialized and are used only rarely.

In general, the greater the number of instructions in an instruction set, the larger the propagation delay is within the CPU. This led some designers to consider eliminating some rarely used instructions from the instruction sets of CPU. They reasoned that reducing the propagation delay within the CPU would allow the CPU to run at a higher frequency, thus performing each instruction more quickly.

However, as in most engineering designs, there is a trade-off. The eliminated instructions generally correspond to specific statements in higher-level languages. Eliminating these instructions from the microprocessor's instruction set would force the CPU to use several instructions instead of one to perform the same function; this would invariably require more time. Depending on how frequently the eliminated instructions were needed, and the number of instructions needed to perform their functions, this approach might or might not improve system performance.

As their names imply，CISC and RISC differ in the complexities of their instruction sets. CISC processors have larger instruction sets that often include some particularly complex instructions. These instructions usually correspond to specific statements in high-level languages. Intel's Pentium-class microprocessors fall into this category. In contrast, RISC processors exclude these instructions, opting for a smaller instruction set with simpler instructions. There are a number of other differences, which we will discuss in the following.

RISC processors have fewer and simpler instructions than CISC processors. As a result, their control units are less complex and easier to design. This allows them to run at higher clock frequencies than CISC processors and reduces the amount of space needed on the processor chip, so designers can use the extra space for additional registers and other components. Simpler control units can also lead to reducing development costs. With a simpler design, it is easier to incorporate parallelism into the control unit of a RISC CPU.

With fewer instructions in their instruction sets, the compilers for RISC processors are less complex than those for CISC processors. As a general guideline, CISC processors were originally designed for assembly language programming, whereas RISC processors are geared toward compiled, high-level language programs. However, the same compiled high-level program will require more instructions for a RISC CPU than for a CISC CPU.

The CISC methodology offers some advantages as well. Although CISC processors are more complex, this complexity does not necessarily increase development costs. Current CISC processors are often the most recent addition to an entire family of processors, such as Intel's Pentium family. As such, they may incorporate portions of the designs of previous processors in their families. This reduces the cost of design and can improve reliability, since the previous design has been proved to work.

CISC processors also provide backward compatibility with other processors in their families. If they are pin compatible, it may be possible simply to replace a previous generation processor with the newest model without changing the rest of the computer's design. This same backward compatibility, whether pin compatible or not, allows the CISC CPU to run the same software as used by the predecessors in its family. For instance, a program that runs successfully on a Pentium Ⅱ should also run successfully on a Pentium Ⅲ. This can translate into significant savings for the user and can determine the success or failure of a microprocessor.

However one assigns RISC and CISC to this analogy, intermingling is currently underway. Newer processor families, such as PowerPC microprocessors, draw some features from RISC methodology and others from CISC, making them a hybrid of RISC and CISC.

Technical Notes to the Text

1. throughput，吞吐量。吞吐量是指在给定的时间周期内，计算机系统能完成的总工作量的一种量度，如每天完成的作业数。

2. shift register，移位寄存器。一种专门执行移位操作的寄存器。

3. parallel processing，并行处理。并行处理是指利用多个功能部件或多个处理机同时工作来提高系统性能或可靠性的计算机系统。

4. SISD (single instruction single data stream)，单指令单数据流。在单指令单数据流计算机中，单个 ALU 对来自控制器的单指令(操作码)流和来自内存的单数据(操作数)流进行操作。

5. SIMD (single instruction multiple data stream)，单指令多数据流。一种带有多个存储器的并行计算机，其中每个存储器都带有一个算术逻辑部件，由一个独立的控制部件按照所需的操作数保存在某处(某个存储器)来分配指令的执行。

6. MISD (multiple instruction single data stream)，多指令单数据流。一种并行处理机，其中多个控制器操作单个数据流。

7. MIMD (multiple instruction multiple data stream)，多指令多数据流。

8. one-dimensional array，一维数组。数组是程序设计语言中由同一属性的多个数据对象组成的聚集，每一数据对象可唯一地通过下标来引用。

9. CISC (complex instruction set computer)，复杂指令集计算机。CISC 是以微程序技术为基础的、具有较复杂指令系统的计算机。

10. RISC (reduced instruction set computer)，精简指令集计算机。RISC 是采用简化了的指令系统和硬连线控制器的计算机，是在高效的流水线技术的基础上充分利用指令并行执行和编译优化技术的计算机。

11. compatibility，兼容性。兼容性是指为一种计算机系统开发的软件或硬件可适用于另一种或其他计算机系统的能力。硬件兼容性是指一种子系统(如内存)或外部设备(如终端)可代替原先指定的设备的能力。软件兼容性是指一台计算机可以直接执行“为另一台计算机用机器语言编写的，或已经过编译汇编的”程序代码的能力。兼容分向下兼容(backward compatibility)和向上兼容(upward compatibility)。

Word Bank to the Text

A. Useful new words

parallel	*adj*. 平行的，类似的，并联的
sequential	*adj*. 连续的，顺序的
concurrent	*adj*. 并发的，协作的，一致的
accomplish	*v*. 完成，达到，实现
feasible	*adj*. 可行的，切实可行的
distinguish	*v*. 区别，辨别
divert	*v*. 转移，转向
manipulate	*v*. (熟练地)操作，使用(机器等)
constitute	*v*. 制定(法律)，建立(政府)，组成
theoretical	*adj*. 理论的
classification	*n*. 分类，分级
segment	*n*. 段，节，片
partition	*n*. 分割，划分
analogous	*adj*. 类似的，相似的
elapse	*v*. (时间)过去，消逝
demonstrate	*v*. 示范，证明
conventional	*adj*. 惯例的，常规的
artificial	*adj*. 人造的，人工的
sophisticated	*adj*. 高度发展的，精密复杂的
utilize	*v*. 利用
matrix	*n*. 矩阵
scalar	*adj*. 梯状的，数量的，标量的
variable	*n*. [数]变数，可变物，变量
paradoxical	*adj*. 反论的，自相矛盾的
propagation	*n*. 繁殖，传播
eliminate	*v*. 排除，消除
invariably	*adv*. 不变地，总是
correspond	*v*. 符合，通信，相应
exclude	*v*. 拒绝接纳，把……排除在外，排斥
compatibility	*n*. 兼容性
intermingle	*v*. 混合
hybrid	*n*. 混血儿，混合物

B. Useful expressions

for the purpose of	为了……目的

at the same time	同时
speed up	加速
distinguish between	区分
separate into	分成
by means of	通过，借助于
under the supervision of	在……监督之下
be visualized as	被想象成
pass through	通过
an industrial assembly line	一条工业流水线
flow through	流过
a stream of numbers	一连串数字
beyond the capabilities of	超出……能力
be characterized by	……的特点在于，……的特点是
represent as	把……描绘成
prior to	在前，居先
incorporate into	合并成
depend on	依靠
as a result	因此
not necessarily	未必

C. Technical terms and proper names

parallel processing	并行处理
serial operation	串行操作
instruction stream	指令流
data stream	数据流
SISD	单指令单数据流
SIMD	单指令多数据流
MISD	多指令单数据流
MIMD	多指令多数据流
pipeline processing	流水线处理
combinational circuit	组合电路
multiplier	乘法器
adder	加法器
clock pulse	时钟脉冲
vector processing	向量处理
one-dimensional array	一维数组
scalar processor	标量处理器
vector instruction	向量指令

CISC	复杂指令集计算机
decoder	译码器
RISC	精简指令集计算机
backward compatibility	向下兼容

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. It is characteristic of ________ that several computations can be in process in distinct segments at the same time.

2. ____________ is made possible by associating a register with each segment in the pipeline.

3. The vector instruction includes the initial address of the operands, ________, and the operation to be performed.

4. ______ processors have fewer and simpler instructions than ________ processors.

5. The sequence of instructions read from memory constitutes an ______. The operations performed on the data in the processor constitutes a __________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. Shift registers operate in serial fashion all the bits of the word at a time. ( )

2. RISC processors have larger instruction sets that often include some particularly complex instructions. ( )

3. SIMD represents an organization that includes many processing units under the supervision of a common control unit. ( )

4. Parallel processing is established by distributing the data among the multiple functional units. ( )

5. RISC is a complex instruction set computer. ( )

Ⅲ. Match each of the following terms with the appropriate definition.

SISD	SIMD	MISD	MIMD

1. ________ Multiple instruction stream, single data stream.
2. ________ Single instruction stream, multiple data stream.
3. ________ Multiple instruction stream, multiple data stream.
4. ________ Single instruction stream, single data stream.

Ⅳ. Translate the following into Chinese.

1. parallel processing 2. pipeline processing 3. vector processing

4. scalar processor　　5. backward compatibility

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

distinguish	correspond	artificial
sequential	variable	feasible
concurrent	eliminate	segment

1. In this way the children are introduced to ________ learning.
2. The twins are so alike that no one can __________ one from the other.
3. She questioned whether it was _________ to stimulate investment in these regions.
4. Lines divided the area into _________.
5. Even in the _________ environment of an office, our body rhythms continue to affect us. Other variables in making forecasts for the industry include the weather and the general economic climate.
6. The Sex Discrimination Act has not _________ discrimination in employment.
7. Academic departments are being _________.
8. The written record of our conversation doesn't _________ with (i.e., is different from) what was actually said.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "perform" and fill in the sentences with the right word.

Example: The register holds the data and the combinational circuit **performs** the **sub-operation** in the particular segment.

1. Three basic pieces of equipment will help you to perform a variety of ex______ to improve your figure.
2. We are the same biologically and our bodies perform the same fu______ everywhere on the planet.
3. Bancroft Community students will live in apartments or in a house, cooking their own meals, washing their own clothes, and learning to perform other ta_______.
4. Valentine ignored the ban and continued to perform ma_______ for young lovers in secret.
5. We were pleased to have the opportunity to watch such good dancers perform a highly praised new ba_______.
6. His council had to perform mi_______ on a tiny budget.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

In computer engineering, computer architecture is the __1__ design and fundamental

operational structure of a computer system. It is a __2__ and functional description of requirements, especially speeds and interconnections, and design implementations for the various parts of a computer — __3__ largely on the way by which the central processing unit (CPU) performs internally and accesses addresses in memory. Computer system architecture __4__ the design of the four __5__ : parallel processing, pipelining, vector processing, and RISC. Parallel processing system is used to provide simultaneous data-processing tasks for the purpose of increasing the __6__ speed of a computer system and is able to perform __7__ data processing to achieve faster execution time. A pipeline is a set of data processing elements connected in series, so that the output of one element is the input of the next one. The elements of a pipeline are often __8__ in parallel or in time-sliced fashion. A vector processor is a CPU design that is able to run mathematical operations on multiple data elements simultaneously. Computers with vector processing capabilities are in demand in __9__ applications. RISC processors have fewer and __10__ instructions than CISC processors. As a result, their control units are less complex and easier to design.

1. A. lastest	B. conceptual	C. ideal	D. simple
2. A. plan	B. design	C. blueprint	D. concept
3. A. relying	B. depending	C. attaching	D. focusing
4. A. refers	B. speaks	C. involves	D. interacts
5. A. ingredients	B. types	C. kinds	D. lists
6. A. computational	B. computer	C. computation	D. computing
7. A. current	B. compound	C. concurrent	D. massive
8. A. displayed	B. carried	C. accomplished	D. executed
9. A. special	B. specialized	C. specific	D. especial
10. A. simple	B. simplifying	C. singular	D. simpler

Translation

Ⅷ. Translate the following into Chinese.

1. The purpose of parallel processing is to speed up the computer processing capability and increase its throughput, that is, the amount of processing that can be accomplished during a given interval of time.

2. It is characteristic of pipelines that several computations can be in process in distinct segments at the same time.

3. To achieve the required level of high performance, it is necessary to utilize the fastest and most reliable hardware and apply innovative procedures from vector and parallel processing techniques.

4. In general, the greater the number of instructions in an instruction set, the larger the propagation delay is within the CPU.

5. Although CISC processors are more complex, this complexity does not necessarily increase development costs.

Chapter 4
Algorithms and Data Structure

Pre-reading Questions

1. What is an algorithm?
2. Can you name at least four of algorithmic techniques?
3. How many classes of data structures do you know? Name at least four of them.

In this chapter, we will discuss about the algorithms and data structure.

4.1 Algorithms

1. The Concept of the Algorithm

An algorithm is an ordered set of unambiguous, executable steps, defining a terminating process. The requirement that an algorithm defines an ordered set of steps means that the steps in an algorithm must have a well-established structure in terms of the order in which its steps are executed. This does not mean that the steps must be executed in a sequence consisting of a first step, followed by a second, and so on. Some algorithms, known as parallel algorithm, for example, contain more than one sequence of steps, each designed to be executed by different processors in a multiprocessor machine. In such cases the overall algorithm does not possess a single thread of steps that conforms to the first-step, second-step scenario. Instead, the algorithm's structure is that of multiple threads that branch and reconnect as different processors perform different parts of the overall task.

Another requirement imposed by the definition is that the steps in an algorithm be unambiguous. This means that during execution of an algorithm, the information in the state of the process must be sufficient to determine uniquely and completely the actions required by each step. In other words, the execution of each step in an algorithm does not require creative skills. Rather, it only requires the ability to follow directions.

The requirement that an algorithm defines a terminating process means that the execution of an algorithm must lead to an end. The origin of this requirement is in theoretical computer science, where the goal is to answer such questions as "What are the ultimate limitations of algorithms and machines?" Here computer science seeks to distinguish between problems whose answers can be obtained algorithmically and lie beyond the capabilities of algorithmic systems. In this context, a line is drawn between processes that culminate with an answer and those that merely proceed forever without producing a result.

Regardless of which side is correct, the fact is that the term *algorithm* is often used in applied, informal settings in reference to sets of steps that do not necessarily define terminating processes.

2. Algorithm Representation

(1) Primitive

The representation of an algorithm requires some form of language. In the case of humans this may be a traditional natural language or perhaps the language of pictures. Often, however, such natural channels of communication lead to misunderstandings, sometimes because the terminology used may have more than one meaning. Problems also arise over misunderstandings regarding the level of detail required. In short, communication problems arise when the language used for an algorithm's representation is not precisely defined or when information is not given in adequate detail.

Computer science approaches these problems by establishing a well-defined set of building blocks from which algorithm representations can be constructed. Such a building block is called a primitive. Assigning precise definitions to these primitives removes many problems of ambiguity and requiring algorithms to be described in terms of these primitives which establishes a uniform level of detail. A collection of primitives along with a collection of rules stating how the primitives can be combined to represent more complex ideas constitutes a programming language.

Each primitive consists of two parts: its syntax and its semantics. Syntax refers to the primitive's symbolic representation, and semantics refer to the concept represented, or the meaning of the primitive. The syntax of air consists of three symbols, whereas the semantics is a gaseous substance surrounding the world.

To obtain a collection of primitives to use in representing algorithms for computer execution, we could turn to the individual instructions that the machine is designed to execute. If an algorithm is expressed at this level of detail, we will certainly have a program suitable for machine execution. However, expressing algorithms at this level is tedious, and so one normally uses a collection of "higher-level" primitives, each being an abstract tool constructed from the lower-level primitives provided in the machine's language. The result is a formal programming language in which algorithms can be expressed in a conceptually higher form than in the actual machine language.

(2) Pseudocode

In general, a pseudocode is a notational system in which ideas can be expressed informally during the algorithm development process. One way to obtain a pseudocode is simply to loosen the rules of the formal language in which the final of the algorithm is to be expressed. This approach is commonly used when the target programming language is known in advance. There the pseudocode used during the early stages of the program development consists of syntax-semantic structure similar to, but less formal than, those used in the target programming language.

The approach to pseudocode is to develop a consistent, concise notation for representing recurring semantic structures. In turn, these structures will become the primitives in which we attempt to express future ideas. For example, the need to select one of two possible activities

depending on the truth or falseness of some condition is a common algorithmic structure. The statements could be rewritten to conform to the structure.

If (condition)　then (activity)

else (activity)

Another common algorithmic structure involves the need to continue executing a statement or sequence of statements as long as some condition remains true. We adopt the following uniform pattern for our pseudocode.

While(condition)　do (activity)

We will often want to refer to values by descriptive names. To make such association, we will use the following form.

assign name the value expression

We want to use our pseudocode to describe activities that can be used as abstract tools in other applications. Computer science has a variety of terms for such program unit, including subprogram, subroutine, procedure, module, and function, each with its own variation of meaning. We will adopt the term *procedure* for our pseudocode and use this term to announce the title by which the pseudocode until will be known. We will begin a pseudocode unit with a statement of the following form.

procedure　name

When the task performed by a procedure is required elsewhere in our pseudocode, we will merely request it by name. For example, if (*condition*) then (*execute the procedure Printout*).

3. Algorithmic Analysis

It is important to recognize the distinction between a problem and an algorithm that solves a problem. A problem has a single problem statement that describes it in some general terms; however, there may be many different ways to solve this problem, and some of these solutions may be more efficient than others. Thus, a number of different algorithms can exist for solving a computational problem, and each of these algorithms could have a different running time complexity. The running time is referring to the computational time required by an algorithm.

(1) Worst-case Analysis

In many algorithms, running time will vary not only for inputs of different sizes, but also for different inputs of the same size. That is, we may find that the running times of an algorithm will vary for inputs of the same size, depending upon the initial ordering of the input data. We define the worst case running time of an algorithm to be the maximum running time of that algorithm over all possible inputs of size n. In addition, it is often found in practical applications that the worst-case running time of an algorithm occurs frequently.

(2) Average-case Analysis

We may also define the average-case running time of an algorithm for an input of size n to be the value that is obtained by averaging the running times of the algorithm over all possible inputs of size n. We will indicate that the average-case running time of an algorithm is being considered

by using "av" as a subscript.

Although average-case analysis may appear to be better than worst-case analysis for measuring running time, usually it is not. Average-case analysis requires that we assume some underlying probability distribution for the input instances. If this assumption is violated in practice, then the average-case analysis may not be meaningful. In addition to the difficulty of determining an appropriate probability distribution function for the input data, such average-case analysis quite often becomes mathematically intractable. Nevertheless, on occasion we will find it convenient to assume that each input instance is equally likely and proceed with an analysis of the running time of that algorithm.

In summary, for the reasons stated above, we will typically use the worst-case measure to express the running time complexity of an algorithm; however, we will use average-case analysis if it makes sense for a particular algorithm or application of an algorithm.

4. Algorithmic Techniques

Below we discuss some general classes of algorithmic techniques. Each of these techniques has special properties that make them appropriate for solving certain types of problems.

(1) Brute-force Algorithm

Brute-force algorithms are characterized by a lack of sophistication in terms of their approach to the solution. They typically take the most direct or obvious route, without attempting to minimize the number of operations required to compute the solution.

Brute-force algorithms are considered quite often in the context of searching. In a searching problem we are required to look through a list of candidates in an attempt to find a desired object. In many cases, the structure of the problem itself allows us to eliminate a large number of the candidates without having to actually search through them. As an analogy, consider the problem of trying to find a frozen pie in an unfamiliar grocery store. You would immediately go to the frozen food aisle, without bothering to look down any of the other aisles. Thus, at the outset of your search, you would eliminate the need to search down most of the aisles in the store. Brute-force algorithms, however, ignore such possibilities and naively search through all candidates in an attempt to find the desired object. This is often called an exhaustive search.

(2) Divide-and-Conquer Algorithm

Divide-and-conquer algorithms solve a problem by breaking it down into several subproblems that are similar to the original problem, except smaller in size. Each of the subproblems is then solved independently. Subsequently, the results obtained from solving these subproblems are combined to produce the solution to the original problem. Because each of the subproblems is similar to the original problem, a recursive approach is typically justified. The steps involved in a divide-and-conquer algorithm can be summarized as follows: First, divide the original problem into a number of subproblems. Next, solve (i.e., conquer) the subproblems. If the subproblems are small enough, they should be solved directly; otherwise, they should be solved recursively. Finally, combine the solutions to the subproblems to produce the final solution.

(3) Dynamic Programming

Algorithms designed using dynamic programming are similar to those developed using divide-and-conquer in that both solve a problem by breaking it down into several subproblems that can be solved recursively. The difference between the two is that in the dynamic programming approach, the results obtained from solving smaller subproblems are reused in the calculation of larger subproblems. Thus, dynamic programming is a bottom-up technique that usually begins by solving the smallest subproblems, saving these results, and then reusing them to solve larger and larger subproblems until the solution to the original problem is obtained. This is in contrast to the divide-and-conquer approach, which solves problems in a top-down fashion. In this case the original problem is solved by breaking it down into increasingly smaller subproblems, and no attempt is made to reuse previous results in the solution of any of the subproblems.

It is important to realize that a dynamic programming approach is only justified if there is some degree of overlap in the subproblems. The underlying idea is to avoid calculating the same result twice. This is usually accomplished by constructing a table in memory, and filling it with known results as they are calculated. These results are then used to solve larger subproblems.

Dynamic programming is often used to solve optimization problems. In an optimization problem, there are typically a large number of possible solutions, and each has a cost associated with it. The goal is to find a solution that has the smallest cost — this is referred to as an optimal solution.

(4) Greedy Algorithm

Another approach that is often used to solve optimization problems is the greedy strategy. Most optimization problems involve a sequence of decisions that must be made appropriately if an optimal solution is to be obtained. In a greedy algorithm, at each decision point，the choice that has the smallest immediate cost is selected, without attempting to look ahead to determine if this choice is part of an optimal solution to the problem as a whole. By locally optimal, we mean a choice that is optimal with respect to some small portion of the total information available about a problem.

(5) Randomized Algorithm

The behavior of a randomized algorithm is dependent not only on the input data, but also on the values produced by a random number generator. If some portion of an algorithm involves choosing between a number of alternatives, and it is difficult to determine the optimal choice, then it is often more efficient to choose a course of action at random rather than taking the time to determine the best alternative. This is particularly true in cases where there are a large number of choices, most of which are“good”.

Although randomizing an algorithm will typically not improve its worst-case running time, it can be used to ensure that no particular input always produces the worst-case behavior. Specifically, because the behavior of a randomized algorithm is determined by a sequence of random numbers, it would be unusual for the algorithm to behave the same way on successive runs —even when it is supplied with the same inputs.

4.2 Data Structure

1. Arrays

Suppose an algorithm for manipulating a series of 24 hourly temperature readings is expressed in a high-level language. A programmer would probably find it convenient to think of these readings arranged as a one-dimensional array called Y whose various entries are referenced in terms of their position in the list. This position is called an index. The first Y might be referenced by Y[1], the second by Y[2], and so on.

The conversion from this conceptual one-dimensional array organization to the actual arrangement within the machine's memory is straightforward. The data can be stored in a sequence of 24 memory cells with consecutive addresses in the same order envisioned by the programmer. Knowing the address of the first cell in this sequence, a translator can convert reference such as Y[5] into the proper memory terminology. It merely subtracts from the index of the desired entry and then adds the result to the address of the memory cell containing the first temperature reading. If the first reading is at address X, the fifth reading is located at address X+(5–l).

Now suppose a programmer wants to write a program dealing with the sales made by a company's sales force during a one-week period. We normally envision such data arranged in tabular form, with the names of the sales personnel listed down the left side and the days of the week listed across top. Hence the programmer would probably like to write the program as though the data were arranged in a two-dimensional array, where the values across each row indicate the sales made by a particular employee, and the values down a column represent all the sales made during a particular day.

A machine's memory is not organized in this rectangular fashion but rather as individual cells with consecutive addresses. Thus the rectangular structure required by the array must be simulated. To do this, we first recognize that the size of the array does not vary as updates are made. We can therefore calculate the amount of storage area needed for the entire array and reserve a block of contiguous memory cells of that size. Next, we store the data in the array row by row. That is, starting at the first cell of the reserved block, we store the values from the first row of the array into consecutive memory locations; following this, we store the next row, then the next, and so on, as shown in Figure 4-1. Such a storage system is said to use row major order in contrast to column major order, in which the array is stored column by column.

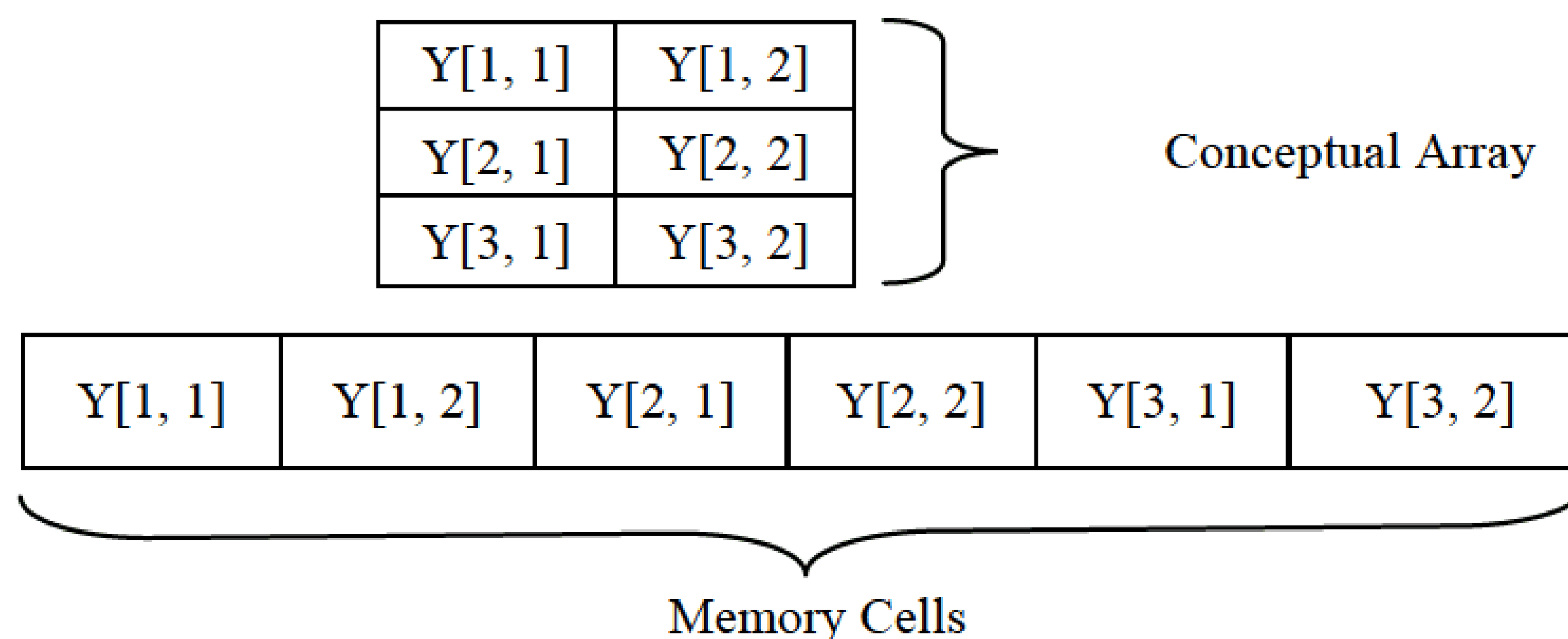


Figure 4-1 A Two-Dimensional Array Stored in Row Major Order

With the data stored in this manner, let's consider how we would find the memory cell containing the value in the third row and second column of the array. We first envision ourselves as being at the first location in the reserved block of the machine's memory. From this location on, we find the data in the first row of the array followed by the second, then the third. To get to the data in the third row, we must move beyond both the first and second rows. Since each row contains two entries, we must move beyond a total of four entries to reach the first entry of the third row. From the beginning of the third row, we must move beyond another one entry to reach the entry in the second column of the array. Altogether, to reach the entry in the third row and second column, we must move beyond 5 entries from the beginning of the block.

The preceding calculation can be refined to obtain a general process that can be used by translators to convert references in terms of row and column positions into actual memory addresses. In particular, if we let c represent the number of columns in an array (which is the number of entries in each row), then the address of the entry in the ith row and jth column will be

$$X+(c\times(i-1)) + (j-1)$$

Where X is the address of the cell containing the entry in the first row and first column. That is, we must move beyond $i-1$ rows, each of which contains c entries, to reach the ith row and then $j-1$ more entries to reach the jth entry in this row. In our prior example $c=2$, $i=3$, and $j=2$, so if the array were stored starting at address X, then the entry in the third row, second column would be address $X+(2\times(3-1))+(2-1)=X+5$.

With this information, software routines can be written to convert requests in terms of row and columns into locations within the block of memory containing the array. A translator, for example, uses this technique to convert a reference such as sales [2, 4] into an actual memory address. A programmer can therefore enjoy the luxury of thinking of the data in tabular form even though it is actually stored in a single row.

2. Lists

An important property of arrays is that their size and shape are constant. Thus, simulating them in a computer's memory is essentially a process of converting the conceptual location of an entry into the actual location. A contrast is provided by dynamic structures that vary in size and shape, such as an organization's membership list, which grows as new members join and shrinks as old members leave. Simulating these conceptual structures requires accommodating variations in the structure itself.

In this section we will consider two ways of implementing the dynamic structure known as a list. But first, we should introduce the concept of a pointer, which is a basic tool used for implementing a variety of dynamic structures.

(1) Pointers

Recall that the various storage locations in a computer's main memory are identified by numeric addresses. If we know the address of a piece of data, we can find that item. Being merely numeric values, these addresses themselves are easily stored in a computer's memory. Thus,

having stored an item of data in one cell of memory, we could store the address of that data in another memory cell and later use this memory cell as a means of retrieving the data. That is, the value in this cell tells us where to find the data.

In a sense, then, a memory cell containing an address can be thought of as pointing to another memory cell. Such cells are called pointers.

Many programming languages allow for the declaration, allocation, and manipulation of pointers, just as they allow for such operations for integers or character strings. Using such a language, a programmer can design elaborate networks of data items within a computer's memory.

(2) List

A list is one of the most fundamental data structures used to store a collection of data items. In fact, lists are used so frequently that it makes sense to define a LIST ADT. The importance of the LIST ADT is that it can be used to implement a wide variety of other ADTs. That is, the LIST ADT often serves as a basic building block in the construction of more complicated ADTs.

A list may be defined as a dynamic ordered n-tuple:

$$L=(l_1,l_2,\ldots,l_n) \tag{4.1}$$

Where l_i is the i-th element in the list. The use of the term dynamic in this definition is meant to emphasize that the elements in this n-tuple may change over time. Notice that these elements have a linear order that is based upon their position in the list. The first element in the list, l_1, is called the head of the list; while the last element, l_n, is referred to as the tail of the list. The number of elements in a list L is denoted by $|L|$, which we will also refer to as the length of the list. Thus the empty list, represented by ( ), has length 0. For the list given in equation (4.1) $|L| = n$.

We will not place any restrictions on the type of elements that a list may store. If all of the elements stored in a list are of the same type, then the list is said to be homogeneous. However, if different types of elements are stored in the list, then the list is said to be heterogeneous. In many applications it is also useful to work with lists of lists. In this case, each element of the list is itself a list. For example, consider the list

$$L=((3), (4, 2, 5), (12, 7, (8, 4), 1), (\,)) \tag{4.2}$$

This list contains four elements. The first element is a list containing a single element 3. The next element is a list containing the elements 4, 2, and 5. The third element in L is a list containing four elements—the first element is 12, the second element is 7, the third element is a list containing the elements 8 and 4, and the last element is 1. The last element in L is an empty list.

The operations we will define for accessing list elements are given below. For each of these operations, L represents a specific list. It is also assumed that a list has a current position variable that refers to some element in the list. This variable can be used to iterate through the elements of a list.

- Insert (L, x, i): Adds element x to L at position i, causing elements l_i, l_{i+1},⋯, l_n to become elements l_{i+1}, l_{i+2},⋯, l_{n+1}, and the length of the list to become n+1. If this operation is successful, the boolean value true is returned; otherwise, the boolean value false is returned.
- Append (L, x): Adds element x to the tail of L, causing the length of the list to become n+1. If this operation is successful, the boolean value true is returned; otherwise, the boolean

value false is returned.

- Retrieve (L, i): Returns the element stored at position i of L, or the null value if position i does not exist.
- Delete (L, i): Deletes the element stored at position i of L, causing elements l_i, l_{i+1},⋯, l_n to become elements l_{i+1}, l_{i+2},⋯, l_{n-1}, and the length of the list becomes n-1. If this operation is successful, the boolean value true is returned; otherwise, the boolean value false is returned.
- Length (L): Returns $|L|$, the length of L.
- Reset (L): Resets the current position in L to the head and returns the value 1. If the list is empty, the value 0 is returned.
- Current (L): Returns the current position in L.
- Next (L): Increments and returns the current position in L. That is, if the current position is i, the current position becomes i+1, and the value i+1 is returned.

Note that only the Insert, Delete, Reset, and Next operations modify the lists to which they are applied. The remaining operations simply query lists in order to obtain information about them. Of course these are only a small number of the operations that can be defined on list. Nevertheless, we will find them quite useful when constructing other ADTs that are based on the LIST ADT.

3. Linked List

Linked list allocates memory for storing list elements as it is needed during run time, and connects list elements together using pointers. Memory is then de-allocated whenever a list element is no longer needed. A linked list is represented by a sequence of nodes connected by links. Because each node in the list is connected to the next by a single link, this data structure is called a singly-linked list. A node in a singly-linked list contains two fields: data, which holds a list element, and next, which stores a link (or pointer) to the next node in the list. The last node in a singly-linked list contains a special symbol that indicates the end of the list. We will assume that this is the null pointer.

A more sophisticated data structure known as a doubly-linked list can also be used to implement the LIST ADT. Each node in a doubly-linked list contains three fields. One field stores a list element, and the other two store links to the preceding and succeeding nodes in the list. Noticing that null pointers are used to mark both ends of the list in this case. But in a circularly-linked list, instead of placing the null pointer in the next field of the tail node, we store a pointer to the head of the list. In addition, dummy nodes which also called sentinels are often added to linked lists. Sentinels can be used to store information about a list, or to simplify the testing of boundary conditions in a list.

The first thing to note about linked lists is that we no longer have direct access to arbitrary elements in the list. In order to access the i-th element, we must traverse the list, beginning at the head, until the desired element is located. Thus, any LIST ADT operation that involves a list position will in the worst case traverse the entire list in order to find the desired position in the

linked list.

In singly-linked lists it is not possible to move backwards to other nodes in the list. This makes it difficult to perform the necessary pointer reassignment during an Insert or Delete operation. For example, we wish to insert a node at position i in a singly-linked list. If we have access to the node at position i–1, then it is a simple matter to "splice" the new node into the list. Similarly, deleting a node from the singly-linked list requires that we modify the next field of the node at position i–1 when we "cut" the appropriate node out of the list. Thus, with singly-linked lists, it is necessary to keep track of the previous node in the list as the traverse associated with an insertion or deletion is being performed.

4. Stacks and Queues

Here we introduce stacks and queues, two special types of dynamic sets that are frequently encountered in computer applications. First, we define the terms FIFO and LIFO. A FIFO is a first-in-first-out structure, which means that the first item stored in this type of structure will be the first item retrieved from it. A LIFO is a last-in-first-out structure. In this case, the last item that is stored in the structure will be the first item retrieved from it.

(1) Stacks

A stack is a dynamic set that obeys the LIFO property. The end of a stack at which elements are inserted and deleted is called the top of the stack. The other end is sometimes called the stack's base. To reflect the fact that access to a stack is restricted to the topmost element, we use special terminology when referring to the insertion and deletion operations. The process of inserting a element on the stack is called a push operation, and the process of deleting a element is called a pop operation.

To implement a stack structure in a computer's memory, it is customary to reserve a block of contiguous memory cells large enough to accommodate the stack as it grows and shrinks. The elements are pushed and popped, the top of the stack moves back and forth within the reserved block of memory cells. A means is therefore needed to maintain a record of the location of the top element. For this purpose, the address of the top element is stored in an additional memory cell known as the stack pointer. That is, the stack pointer points to the top of the stack.

To push a new element on the stack, we first adjust the stack pointer to point to the vacancy just beyond the top of the stack and then place the new element at this location. To pop an element from the stack, we read the data pointed to by the stack pointer and then adjust the pointer to point to the next element down on the stack.

A classic application of stack involves the execution of a program involving procedures. A procedure is called, a pointer to the pertinent return location is pushed on a stack, and as a procedure is completed, the top element from the stack is extracted with the assurance of obtaining a pointer to the proper return location.

(2) Queues

A queue is a dynamic set that obeys the FIFO property. The concept of a queue is inherent in

any system in which objects are served in the same order in which they arrive. The ends of a queue get their names from this waiting-line relationship. The end at which elements are removed is called the head of the queue just as we say that the next person to be served in a cafeteria is at the head of the line. Similarly, the end of the queue at which new elements are added is called the tail.

We can implement a queue in a computer's memory within a block of contiguous cells. Since we need to perform operations at both ends of the structure, we set aside two memory cells to use as pointers instead of just one. One of these pointers, called the head pointer, keeps track of the head of the queue; the other, called the tail pointer, keeps track of the tail. When the queue is empty, both of these pointers point to the same location. Each time an element is inserted, it is placed in the location pointed to by the tail pointer and then the tail pointer is adjusted to point toward the next unused location. In the manner, the tail pointer is always pointing to the first vacancy at the tail of the queue. Removing an element from the queue involves extracting the element pointed to by the head pointer and then adjusting the head pointer to point toward the element that followed the removed element.

5. Trees

(1) The Concept of Trees

A tree is a finite nonempty set of elements. One of these elements is called the root, and the remaining elements are partitioned into trees which are called the subtrees of t.

When drawing a tree, each element is represented as a node. The tree root is drawn at the top, and its subtrees are drawn below. Each subtree is drawn similarly with its root at the top and its subtrees below. At times we refer to tree structures as though each node gives birth to those nodes immediately below it. In this sense, we often speak of a node's ancestors or descendants. We refer to its immediate descendants as its children and its immediate ancestor as its parent. Moreover, we speak of nodes with the same parent as being twins or siblings. In a tree, nodes with no children are called leaves. Another commonly used tree term is level. The tree root is at level 1, its children are at level 2, their children are at level 3, and so on. The degree of a node is the number of children it has. The degree of a leaf is zero. The degree of a tree is the maximum of its node degrees.

(2) Binary Tree

A binary tree T is a finite collection of elements. When the binary tree is not empty, it has a root element and the remaining elements are partitioned into two binary trees, which are called the left and right subtrees of T.

The essential differences between a binary tree and a tree are: (1) A binary tree can be empty, whereas a tree cannot; (2) Each node in a binary tree has exactly two subtrees (one or both of these subtrees may be empty), which in a tree can have any number of subtrees; (3) The subtrees of each node in a binary tree are ordered. While the subtrees in a tree are unordered.

The depth of a binary tree is the number of levels in it. A binary tree of depth h, $h \geqslant 0$, has at least h and at most 2^h-1 notes in it. The purpose of discussing tree storage techniques is to restrict our attention to binary trees, which are trees in which each node has at most two children. Such

trees are normally stored in memory using a linked structure similar to that of linked lists. Each node of binary tree contains three components: the data, a pointer to the node's first child, and a pointer to the node's second child. Although there is no left or right inside a machine, it is helpful to refer to the first pointer as the left child pointer and the other pointer as the right child pointer in reference to the way we would draw the tree on paper.

Storing the tree in memory involves finding available blocks of memory cells to hold the nodes and linking these nodes according to the desired tree structure. That is, each pointer must be set to point to the left or right child of pertinent node or assigned the NIL value if there are no more nodes in that direction of the tree. This means that a terminal node is characterized by having both of its pointers assigned NIL. Finally we set aside a special memory location, called a root pointer, where we store the address of the root node. It is root pointer that provides initial access to the tree.

An alternative to a linked storage system for binary trees is the technique of setting aside a contiguous block of memory cells, storing the root node in the first of these cells, storing the left child of the root in the second cell, storing the right child of the root in the third cell, and in general, storing the left and right children of the node found in cell n in the cells $2n$ and $2n+1$, respectively. Cells within the block that represent locations not used by the current tree structure are marked with a unique bit pattern that indicates the absence of data.

Some of the operations that are commonly performed on binary trees are to: (1) Determine its height; (2) Determine the number of elements in it; (3) Make a copy; (4) Display the binary tree on a screen or on paper; (5) Determine whether two binary trees are identical; (6) Delete the tree; (7)If it is an expression tree, evaluate the expression; and (8) If it is an expression tree, obtain the parenthesized form of the expression.

All these operations can be performed by traversing the binary tree in a systematic manner. In a binary tree traversal, each element is visited exactly once. During this visit, all action with respect to this element is taken. This action can include writing the element on a screen or on paper, evaluating the expression represented by the subtree of which this element is the root, adding one to a running count of the number of elements in the binary tree, and deleting the node used by this element.

Technical Notes to the Text

1. algorithm，算法。算法指在有限步骤内求解某一问题所使用的一组定义明确的规则，如为达到给定精度计算 sin$x$ 的全部运算过程的完整说明。

2. pseudocode，伪码。伪码是一种非正式的、灵活的程序设计语言，但它在计算机上不能直接运行，只是程序员在编码之前组织、表达其设计思想的主要工具。

3. syntax，语法。程序设计语言的语法是指程序的组成规则。

4. semantics，语义。程序设计语言的语义是指程序的含义。

5. singly-linked list，单向链表。单向链表中每个结点(除表中最后一个结点外)都有且只有一个后继结点，其后继结点由该结点的指针指出。

6. circularly-linked list，循环链表。循环链表的最后一个结点的指针指向的是第一个结点的地址。

7. FIFO (first in/first out)，先进先出。一种排队方法，按此方法，下一次要被检索出的项将是在队列中等待时间最长的项。

8. LIFO (last in/first out)，后进先出。一种排队方法，按此方法，下一个要取出的项是最近进入该队列的项。

9. stack，栈。栈是只能在始端进行插入、删除操作的线性表，是一种后进先出型的结构。

10. queue，队列。队列是一种线性表，只能在始端删除结点，在终端插入结点，是一种先进先出的结构。

11. tree，树。树是连通无回路的图。

12. root，根。树中唯一没有父结点的结点称为根。

13. binary tree，二叉树。二叉树是一棵只有左右两个子树的树。

Word Bank to the Text

A. Useful new words

algorithm	*n.* 算法
unambiguous	*adj.* 不含糊的，明确的
conform	*v.* 使一致，使遵守
scenario	*n.* (行动的)方案，游戏的关，或是某一特定情节
culminate	*v.* 达到顶点
terminology	*n.* 术语学
primitive	*n.* 基元机器指令
syntax	*n.* 句法，句子结构学
semantics	*n.* 语义学
tedious	*adj.* 单调乏味的，沉闷的
pseudocode	*n.* 伪代码
notation	*n.* 符号
uniform	*adj.* 统一的，相同的，一致的
distinction	*n.* 区别，差别
underlying	*adj.* 基础的，基本的
violate	*v.* 违犯，干扰，违反
intractable	*adj.* 难处理的
naively	*adv.* 轻信地，缺乏经验或判断力地
exhaustive	*adj.* 无遗漏的，彻底的，详尽的
recursively	*adv.* 递归地
optimal	*adj.* 最佳的，最理想的

randomize	*v*. 使随机化
dimensional	*adj*. 维量[数]的，……维的
straightforward	*adj*. 简单的，直截了当的
consecutive	*adj*. 连续的，连贯的
envision	*v*. 想象，预想
subtract	*v*. (常与 from 连用)减去，扣除
simulate	*v*. 模拟，模仿
contiguous	*adj*. 邻近的，接近的
preceding	*adj*. 在前的，前述的
routine	*n*. 常规，程序
accommodate	*v*. 供应，供给
retrieve	*v*. 寻回，取回
allocation	*n*. 分配，安置
elaborate	*adj*. 精心制作的，详细阐述的
homogeneous	*adj*. 同类的，相似的
iterate	*v*. 反复说，重申
dummy	*adj*. 虚拟的，假的，虚构的
vacancy	*n*. 空，空白，空缺，空闲
extract	*v*. 拔出，拉出
sentinel	*n*. (表示某段信息开始或终了的)标记
traverse	*v*. 通过，经过，横过

B. Useful expressions

in a sequence	按次序
in other words	换句话说
lead to	导致，通向
regardless of	不管，不顾
in reference to	关于
in advance	预先
conform to	符合，遵照
in practice	在实践中，实际上
in summary	总之
make sense	有意义
a lack of	缺乏
be involved in	涉及，专心
in contrast to	和……形成对比
make an attempt to	企图，尝试
look ahead	向前看，展望未来

as a whole	总体上
be dependent on	依靠，依赖
at random	随机
supply with	供应
convert into	转化成
subtract from	减去
as though	好像，仿佛
envision as	想象为
in particular	特别
in a sense	在某种意义上
be thought of as	被认为是
have access to	有权使用
retrieve from	重新得到
adjust to	适应，调节
set aside	留出
keep track of	纪录
give birth to	产生
partition into	分割，划分
with respect to	关于，至于

C. Technical terms and proper names

algorithm	算法
parallel algorithm	并行算法
primitive	原语
syntax	语法
semantics	语义
pseudocode	伪码
exhaustive search	穷举搜索
divide-and-conquer algorithm	分治算法
dynamic programming	动态规划
bottom-up	自下而上
top-down	自上而下
array	数组
one-dimensional array	一维数组
two-dimensional array	二维数组
pointer	指针
program counter	程序计数器
instruction pointer	指令指针
list	列表

linked list	链表
singly-linked list	单向链表
doubly-linked list	双向链表
circularly-linked list	循环链表
FIFO	先进先出
LIFO	后进先出
stack	栈
push	压栈
pop	出栈
stack pointer	栈指针
queue	队列
tree	树
root	根
level	层次
degree of a node	结点的度
depth of a tree	树的深度
binary tree	二叉树
traversal	遍历
M-way search tree	M 向搜索树

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. Each primitive consists of two parts: ________ and ________.

2. A ________ is a notational system in which ideas can be expressed informally during the algorithm development process.

3. The ________ refers to the computational time required by an algorithm.

4. We define the worst case running time of an algorithm to be the _______ running time of that algorithm over all possible inputs of size *n*.

5. A ______ is one of the most fundamental data structures used to store a collection of data items.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. A number of different algorithms can exist for solving a computational problem, and each of these algorithms could have a same running time complexity. (   )

2. The divide-and-conquer algorithm is a bottom-up technique that usually begins by solving the smallest subproblems, and the dynamic programming solves problems in a top-down fashion. ( )

3. An important property of arrays is that their size and shape are constant. ( )

4. If all of the elements stored in a list are of the same type, then the list is said to be heterogeneous. However, if different types of elements are stored in the list, then the list is said to be homogeneous. ( )

5. A queue is a dynamic set that obeys the LIFO property. ( )

Ⅲ. Answer the following questions.

1. List the steps involved in a divide-and-conquer algorithm.
2. List approach that is often used to solve optimization problems.
3. Describe the differences between a binary tree and a tree.

Ⅳ. Translate the following into Chinese.

1. parallel algorithm 2. exhaustive search 3. dynamic programming
4. doubly-linked list 5. two-dimensional array

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

conform	exhaustive	retrieve
uniform	elaborate	violate
tedious	envision	emerge

1. Some people do not ________ to academic expectations or the social abilities of others, and they are placed outside society.
2. All flowing water, though it appears to be ________, is actually divided into extensive inner surfaces, or layers, moving against one another.
3. This is by no means an _______ list but it gives an indication of the many projects taking place.
4. Most people do stop at this point, not ________ that there is anything beyond.
5. The company has worked out an ________ management training scheme for graduates.
6. The men were trying to ________ weapons left when the army abandoned the island.
7. There is growing evidence that the economy is at last ________ from recession.
8. They went to prison because they ________ the state law.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can “satisfy” and fill in the sentences with the right word.

Example: A binary search tree is a binary tree that may be empty. A nonempty binary search

tree **satisfies** the following **properties**.

1. The energy in the heat released by a hurricane's rainfall in a single day would satisfy the entire electrical ne_______ of the United States for more than six months.

2. The teacher had the pupils quizzing each other to satisfy their surplus en______.

3. During the adventures, they found some edible wild berries enough to satisfy their hu______.

4. Advertisements sell more than products: they sell human hopes and dr______. But buying a product can never satisfy hopes and dreams.

5. The firm is unable to satisfy all the de______ of customers for its new car this year.

6. Some books satisfy the reader's cu______ about life in other lands.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

Algorithms are essential to the way computers process information, because a computer program is __1__ an algorithm that tells the computer what specific __2__ to perform in certain order in order to carry out a specified task, such as calculating employees' paychecks or printing students' report cards. Thus, an algorithm can be considered to be __3__ sequence of operations that can be performed by a turning-complete system. In fact, a data structure is a way of __4__ data in a computer so that it can be used efficiently. Often a carefully __5__ data structure will allow the __6__ efficient algorithm to be used. The choice of the data structure often begins from the choice of a(n) __7__ data type. A well-designed data structure allows a variety of critical operations to be performed, using as few resources, both execution time and memory space, as possible. Data structures are implemented by a(n) __8__ language as data types and the references and operations they provide. Moreover, different kinds of data structures are __9__ to different kinds of applications, and some are highly specialized to certain tasks. For example, B-trees are particularly well-suited for implementation of databases, __10__ networks of machines rely on routing tables to function.

1. A. essentially	B. essential	C. partially	D. mainly
2. A. methods	B. steps	C. plans	D. exercises
3. A. some	B. a	C. the	D. any
4. A. receiving	B. storing	C. input	D. output
5. A. chosen	B. made	C. programmed	D. picked
6. A. very	B. most	C. more	D. less
7. A. abstract	B.concrete	C. special	D. certain
8. A. basic	B. running	C. programming	D. advanced
9. A. comforted	B. suited	C. compared	D. hooked
10. A. therefore	B. with	C. when	D. while

Translation

Ⅷ. Translate the following into Chinese.

1. In short, communication problems arise when the language used for an algorithm's representation is not precisely defined or when information is not given in adequate detail.

2. Another common algorithmic structure involves that the need to continue executing a statement or sequence of statements as long as some condition remains true.

3. In many algorithms, running time will vary not only for inputs of different sizes, but also for different inputs of the same size.

4. Thus, dynamic programming is a bottom-up technique that usually begins by solving the smallest subproblems, saving these results, and then reusing them to solve larger and larger subproblems until the solution to the original problem is obtained.

5. The conversion from this conceptual one-dimensional array organization to the actual arrangement within the machine's memory is straightforward, and the data can be stored in a sequence of 24 memory cells with consecutive addresses in the same order envisioned by the programmer.

Chapter 5
Programming and Languages

Pre-reading Questions

1. What is a program?

2. Name at least four of the programming languages.

3. How many important terminologies of object-oriented programming do you know? Name at least four of them.

A total computer system includes both hardware system and software system. Hardware consists of the physical components and all associated equipment. Software refers to the programs that are written for the computer. It is possible to be familiar with various aspects of computer software without being concerned with details of how the computer hardware operates. It is also possible to design parts of the hardware without knowledge of its capabilities. However, those concerned with computer architecture should have knowledge of both hardware and software because the two branches influence each other.

In this chapter, we will introduce the procedure of programming, the development of programming languages, compiling and assembling programs, object-oriented programming, visual programming, and Internet programming.

5.1 The Procedure of Programming

A program is a list of instructions that the computer must follow to process data into information. The instructions consist of statements written in a programming language. Programming, also called software engineering, is a multistep process for creating that list of instructions. Programming is traditionally a five-step problem-solving process. The five steps are as follows:

Step 1: Problem definition. This step requires clarifying objectives, output, input, and processing tasks, and studying their feasibility and finally documenting them.

Step 2: Program design. This step includes determining program logic, designing details using pseudocode and/or flowcharts (preferably with control structures), and testing design with structured walkthrough. For example, structure programming is designed in three mini-steps. First, the program logic is determined through a top-down approach, on the basis of program modules and a hierarchy chart. Then it is designed in detailed form, or in narrative form, using pseudocode, or graphically, using flowcharts. Finally, the design is tested with a structured walkthrough.

Step 3: Program coding. The writing of the program is called coding. Coding consists of

translating the logic requirements from pseudocode or flowcharts into a programming language — the letters, numbers, and symbols arranged according to syntax rules that make up the program. Coding is what many people think of when they think of programming, although it is only one of the five steps.

Step 4: Program testing. Program testing involves running various tests — such as desk-checking and debugging — and then running actual data to make sure the program works.

Step 5: Program documentation and maintenance. Documentation is needed for everyone who will be involved in the program — users, operators, and programmers. So documentation is classified into user documentation, operator documentation, and programmer documentation. User documentation should explain both how to use the software and how to exploit it. It should also include information on what a user should do on encountering an error. The operator documentation provide information to the operator. The information includes what to do when the program malfunctions. The programmer documentation offers the keys to understanding the program's underlying logic and operation. Preparing documentation is not just an end-step process of programming. It has been going on throughout all previous programming steps. Maintenance is any activity designed to keep programs in working condition, error-free, and up to date. It includes adjustments, replacements, repairs, measurements, tests, and so on.

5.2 The Development of Programming Languages

A program is a list of instructions or statements for directing the computer to perform a required data processing task. There are various types of programming languages that one may write for a computer. Traditionally, computer programming languages are divided into five levels or generations, which are as follows: (1) First generation — machine language, (2) Second generation — assembly language, (3) Third generation — high-level language, (4) Fourth generation — very-high-level language, (5) Fifth generation — natural language.

1. Machine Language

Machine language is the lowest level of programming languages. The language contains the binary values that cause the microprocessor to perform certain operations. When a microprocessor reads and executes an instruction, it is a machine language instruction. The computer hardware recognizes only this type of instruction. However, programmers looking at this program will have a difficult time understanding what is to be achieved when this program is executed. So programmers do not write programs in machine language. Rather, programs written in a high-level language or assembly language are converted to machine language, which is then executed by the computer. Machine languages are platform-specific, because each microprocessor has its own machine language of its predecessor processors.

2. Assembly Language

Assembly languages are at a much lower level of abstraction. As with machine language, each

microprocessor has its own assembly language. A program written in the assembly language of one microprocessor cannot be run on a computer that has a different microprocessor — with one very important exception. A company that develops microprocessors will usually design new processors so that they are backward compatible with its previous microprocessors. For example, Intel's Pentium Ⅲ microprocessor can run programs written in the assembly languages of its Pentium Ⅱ, Pentium Pro, Pentium, 80486, 80386, 80286, and 8086 microprocessors — basically any code ever written for an IBM-compatible personal computer. With backward compatibility, someone can buy a new computer with a state-of-the-art microprocessor and still use the same software that they used on their old computer. Such cost and time saving are an important marketing consideration in microprocessor and assembly language instruction set design.

Unlike high-level languages, instructions in assembly languages can directly manipulate the data stored in a microprocessor's internal components. Assembly language instructions can load data from memory into a microprocessor's registers, add values, and perform many other operations. Assembly languages are not platform-independent.

3. High-Level Language

High-level languages hide the details of the computer and operating system on which they will run from the programmer. They are said to be platform-independent, the same program code can be converted and run on computers with different microprocessors and operating systems without modification. Languages such as C, BASIC, and FORTRAN are high-level languages.

High-level languages are also known as procedural languages. The corresponding programs set forth precise procedures, or series of instructions and the programmer has to follow a proper order of actions to solve a problem. To do that, the programmer has to have a detailed knowledge of programming and of the computer that the program will run on. Suppose you want to take a taxi to a theater showing a particular movie. If you tell the taxi driver precisely how to get to the theater, that's procedural. You have to know how to get there yourself, and you will probably get there efficiently. However, if you simply tell the taxi driver to "take me to see movie X", then you're saying only what you want, which is nonprocedural. In this case you may not get to the theater in an efficient manner.

4. Very-High-Level Language

Very-high-level languages are often called 4GLs, for fourth-generation languages compared with third-generation languages. 4GLs are much more user-oriented and allow programmers to develop programs with fewer commands, although they also require more computing power. 4GLs are called nonprocedural because programmers and even users can write programs that only tell the computer what they want to do, without specifying all the procedures for doing it. That is, they do not have to specify all the programming logic or otherwise tell the computer how the task should be carried out. This saves programmers a lot of time because they do not need to write as many lines of code as they do with procedural languages. 4GLs are also called RAD (rapid application development) tools.

Very-high-level languages consist of report generators, query languages, application generators, and interactive database management system programs. Some 4GLs are tools for end-users, and some are tools for programmers.

4GLs may not entirely replace third-generation languages because they are usually focused on specific tasks and hence offer fewer options. Still, they improve productivity because programs are easy to write.

5. Natural Language

Natural languages are of two types. The first is ordinary human language: English, Spanish, and so on. The second is programming language using human language to give people a more natural connection with computers. Some of the query languages mentioned above under 4GLs might seem pretty close to human communication, but natural languages try to be even closer.

With 4GLs, you can type in some rather routine inquiries. Natural languages allow questions or commands to be framed in a more conversational way or in alternative forms.

Natural languages are part of the field of study known as artificial intelligence (AI). Artificial intelligence is a group of related technologies that attempt to develop machines to emulate human-like qualities.

5.3 Compiling and Assembling Programs

Once a programmer has written a program in a high-level or assembly language, the program must be converted to machine code. High-level language programs are compiled, and assembly language programs are assembled. We now look at these processes from a high-level perspective. We examine the processes themselves, but not the internal workings of compilers and assemblers.

A program written in a high-level language is input to a compiler. The compiler checks to make sure every statement in the program is valid. When a program has no syntax errors, the compiler finishes compiling the program, the source code, and generates an object code file. The object code is the machine language equivalent of the source code.

At this point the program has been compiled successfully, but it is not yet ready to be executed. Some programs use the object code of other programs in addition to their own. A linker combines your object code with any other required object code. This combined code is stored as an executable file. It is actually the code in this file that the computer runs. A loader copies the executable file into memory; the microprocessor then runs the machine code contained in that file.

As noted earlier, high-level languages are platform-independent. The same high-level source code can be compiled to run on different microprocessors and operating systems, or computing platforms. In practice, each platform has a separate compiler, although it is theoretically possible to have a single compiler that produces different object code for different platforms.

A high-level language statement is usually converted to a sequence of several machine code instructions. Furthermore, there may be more than one valid conversion for a statement, which can

complicate the design of a compiler. This is not the case with assembly language. Each assembly language instruction corresponds to one unique machine code instruction. For this reason, assemblers are much less complex than compilers.

The assembler, like the compiler, converts its source code to object code. From there, it follows the linking and loading procedure that was used for compiled code. Each assembly language is specific to one microprocessor; we don't need assemblers for different platforms because assembly language programs usually run on only one platform.

5.4 Object-Oriented Programming (OOP)

Object-oriented programming (OOP) is a programming method that combines data and instructions for processing that data into a self-sufficient "object" that can be used in other programs. The important thing here is the object. Figure 5-1 shows the organization of information in the classes and objects of an object-oriented programming.

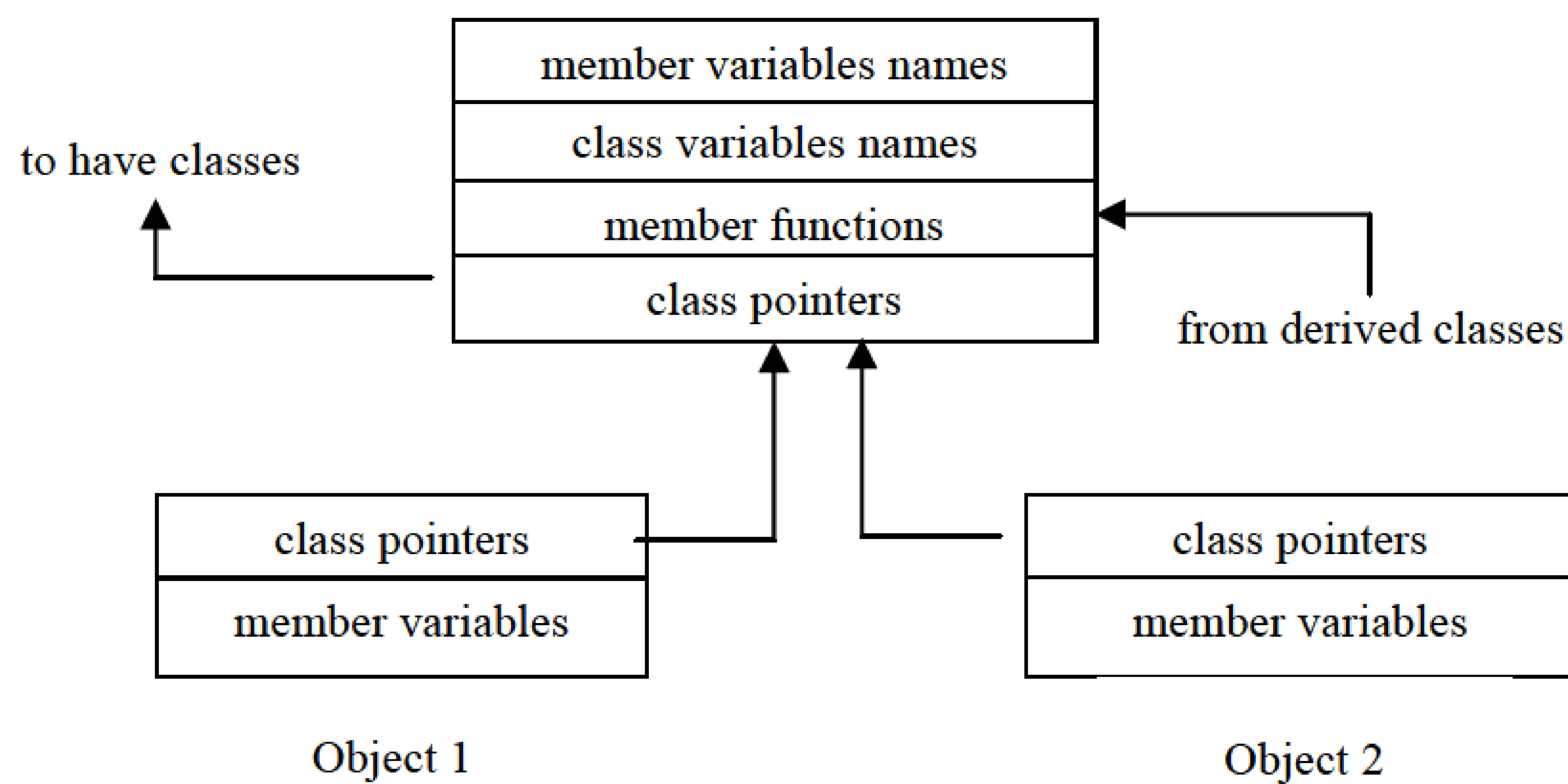


Figure 5-1 Class and Objec.

1. Classes and Objects

Computation in an object-oriented system involves manipulating class objects. A class is actually a means of packaging an ADT (abstract data type). That is, the class concept allows the data elements as well as the accessing routines of an ADT implementation to be encapsulated together as a single entity. Furthermore, the class controls the accessing of the data elements. An application program can only access the data elements in the class through the specified interface provided by the accessing routines. This effectively constructs the wall that should exist between an application program and the implementation level of a data type. A class contains all the information necessary to construct separate instances of itself; these instances are referred to as objects. If the objects are chosen to represent objects in the real world, a more natural mapping of the problem domain to the software domain can be obtained.

It is important to understand the difference between a class and an object of that class. A class

is simply a specification for creating objects. Thus, a single class may create multiple objects. As mentioned above, the objects are the actual entities that will be manipulated in a computer program. This is analogous to the difference between a data type and an instance of that data type. Multiple instances of a data type may be created in a program. These instances, not the data type itself, are then manipulated in the computer program.

Every object contains its own set of data elements, called member variables or data members, and determines the individual state of that object. In addition, a class may store information that is shared among all instances of the class in class variables. Member and class variables are packaged in such a way that they may only be accessed through the accessing routines provided by the class. The accessing routines of a given class are shared by all objects of that particular class. In the terminology of object-oriented programming, these accessing routines are called member functions.

All instances (i.e., objects) of a class point to the same set of member functions, but each object keeps its own copy of the member variables.

2. Inheritance

Inheritance is the means by which objects of a class can access member variables and functions contained in a previously defined class, without having to restate those definitions. This will give us the ability to create a new class that is an extension or specialization of an existing class. In this case, the new class (i.e., the derived class) is said to be derived from the existing base class. Object-oriented programming languages must also support multiple inheritance in which a derived class can inherit more than one base class.

A derived class inherits member variables, class variables, and member functions from its base class. The derived class may also add new member variables, class variables, or member functions that are necessary for its specialized operations. Additionally, a derived class may redefine any member function provided by the base class by simply supplying a new member function that has the same name as the old member function in the base class. In this case, the new member function in the derived class is said to overload the member function with the corresponding name in the base class. This allows different meanings to be attached to the same member function name, which member function is invoked when the name called in a program depends upon the specific class of the object being used.

The motivation for inheritance follows from the observation that in many cases new software development evolves from previous development efforts. This fact, largely ignored in classical software design, is of central concern in object-oriented design. Using inheritance, programmers are able to avoid the time consuming and error-prone task of rewriting large amounts of code that perform nearly the same tasks as some existing piece of code. Therefore, through the use of inheritance, an addition to a system often requires just that addition and not modification.

The relationship among classes in an object-oriented program is often depicted as an inheritance graph. This structure pictorially represents the hierarchical organization of the classes.

For example, the inheritance graph of a hierarchy of classes that can be developed for creating and manipulating graphical objects on a computer screen. The base class in this hierarchy, shape, is defined to represent the general concept of a shape. Specific shapes are then derived from shape. Furthermore, these specific shapes can be manipulated using the interface provided by the shape class. Notice that more general shapes appear towards the top, and more specific shapes appear towards the bottom of the inheritance graph. By organizing shapes in this manner, we can write computer programs that manipulate general shapes without having to worry about the specific type of the shape. This means that additional shapes can be created and added through inheritance, and that previously developed code will not have to be modified in order to use these shapes.

3. Message Passing

It is often useful to think of computation in an object-oriented system as evolving around messages. Using this analogy, objects in the system manipulate other objects by sending messages requesting them to perform specific actions. These messages invoke the appropriate member functions in the object's classes, possibly causing the objects themselves to change state. If a desired member function is not found in an object's immediate class, then member functions in that object's base class are searched. If it is not found there, the search continues on up the hierarchy of the object's inherited classes until either the appropriate member function is found, or the class at the root of the hierarchy is reached. If the desired member function is not found in the root class, then an error has occurred.

There is a fundamental distinction between this message passing approach and the conventional procedure calls used in procedure-oriented languages. A message can be viewed as a request of an object to perform some action. How the object responds to this request depends upon available member functions. This approach allows objects from different classes to respond appropriately to the same messages. In addition, a new object can easily replace an existing object in a software system if it responds to the same messages as the existing object. On the other hand, a procedure call specifies how an action should be accomplished, not what action is being requested.

4. Dynamic Binding and Polymorphism

When a software system determines how a particular operation is to be performed on an object, it is said to bind a specific implementation of that operation to the object. If the system decides on which implementation of the operation to use at compile time, it performs static binding. If this choice is made at run time, then the system performs dynamic binding. That is, in static binding, the static form of an object (i.e., its form at compile time) determines which version of an operation will be used. In dynamic binding, the dynamic form of an object (i.e., its form during run time, when an operation is needed) determines which version of an operation is used.

An object-oriented programming language typically supports static binding, but it must support dynamic binding. This gives operations (i.e., messages) the important capability of being able to automatically adapt to the objects to which they are applied.

Another important feature of object-oriented languages that is directly supported by dynamic binding is polymorphism. Polymorphism is defined as the ability to assume various forms. In object-oriented programming this term is used to denote the ability of a single message to refer at run time to objects of different classes. Typically a polymorphic function is declared in a base class. Then, this same function is redefined in classes that are derived from the base class (i.e., the function name is overloaded). Thus, functions with the same name as the one in the base class exist in the derived classes. Now, if an object of the base class is declared in a program, the original function definition found in the base class will be invoked when the function is called. However, if an object of a derived class is later assigned to the base class object, then the function definition for the derived class will be invoked if the same function is called.

If the function in the base class was not declared as polymorphic, then static binding is performed, and the type (i.e., class) of an object at compile time determines which function definition is invoked. Thus, in the example given above, the original type of the object was that of the base class. Therefore, even if objects of different types were assigned to this object, it would still invoke the function definition found in the base class whenever the function was called. However, when the function is declared to be polymorphic, dynamic binding occurs, and the software system will check the current type of the object before invoking the appropriate function definition.

5. Putting It All Together

Taken separately, the issues discussed so far are important, but in combination they have a profound influence on the way software can be developed and used. The most significant contribution of the object-oriented paradigm involves the increase in reusability of system components; this is supported through the use of data encapsulation, message passing, polymorphism, and inheritance.

Because the data in an object can only be accessed through a well-defined external interface (i.e., the member functions) and these interface routines are packaged with the data, the implementation of the class is effectively “hidden” from the rest of the system. Thus, the class concept supports data encapsulation. Notice also that a class directly supports the implementation of ADTs. The abstract view of the data type in this case is the view provided by the member functions that manipulate the data. The ability to use data abstraction is fundamental to any design approach. Therefore, the data encapsulation and data abstraction provided by object-oriented languages allows us to develop ADTs that perform well-defined sets of operations, independent of a particular application. These ADTs may then be used as necessary in applications that require their functionality.

Additionally, through the use of inheritance, the design process may be simplified and development time reduced by adding functionality to previously developed classes. Most importantly, this additional functionality can be added to the existing class without modification to the class itself. Therefore, the consumer of the class does not need to be concerned with the implementation details of the class. The consumer only needs to be familiar with the functionality provided by the external member functions.

The significance of the previous point cannot be overemphasized. Using traditional software technology, the supplier is responsible for system modification. By allowing the objects to determine how a message should be interpreted, the responsibility of implementing system modifications and additions can be shifted away from the supplier of a class to the consumers of a class. Such a fundamental change in focus supports the notion that software can be developed and packaged for later use just as hardware components are packaged for convenient use in integrated circuits (ICs). That is, class suppliers can serve the purpose of developing, debugging, and testing a set of base classes — just as corresponding IC manufacturers develop, test, and debug their hardware products. These base classes (or software ICs) are then made available to consumers for possible inclusion into their software systems.

The benefits obtained from such an approach are the same benefits obtained by hardware engineers through the use of ICs. The design and development of a system does not have to proceed through the lower levels of abstraction because fully tested sub-system components are already available. Therefore, the designers do not have to concern themselves with the detailed working of these sub-system components. Instead of starting every new development from scratch, the object-oriented approach allows the developer to use a previous working implementation that has already been thoroughly tested. The developer may then use inheritance to provide only those changes that are necessary to meet the new design specifications.

It should also be noted that these same benefits are obtained if the software components are developed and used by the same person. (i.e., the supplier is the consumer.) The design of applications can be simplified and the development time reduced because the developer of a class is still not required to deal with the complexities of its implementation when using the class. This means that the developer does not have to recall the implementation details of the class in order to modify and reuse the class in other applications.

6. Design Philosophy

Object-oriented program design is best accomplished by analyzing the classes of objects in the physical system. This design approach consists of a successive refinement of the descriptions and interactions of these objects. Next, the collection of objects in the physical system is modeled using classes in the software system. The class structure offered by object-oriented programming languages supports this approach by allowing objects in the physical system to be modeled as the same objects in the software system. In addition, these classes can be easily extended or combined, enabling them to form more complex components. Existing classes are used whenever possible. If new classes need to be included, they are developed in a generic manner so that their services might be used in other systems. Finally, the inheritance mechanism leaves the system open for future modification or incorporation into other systems as needed, thereby placing extendibility and reusability at the heart of object-oriented design.

One of the most difficult aspects of object-oriented design involves determining the organization of classes in the software system architecture. There are many different ways in

which classes and objects can interact. Defining relationships between the entities that may exist in an object-oriented system often makes it easier to understand, discuss, and modify the system. We will consider the following relationships:

- IS-A relationship: This is a specialization relationship which is used to indicate that one class is a variant of another class. Stating that “class B IS-A class A” indicates that the major characteristics of class B are inherited from class A.
- HAS-A relationship: This is a containment relationship which is used to indicate that one class or object is part of some other class or object. While IS-A can only be used to define a relationship between classes, HAS-A can be used to define a relationship between classes, between an object and a class, or between objects. Stating that “class B HAS-A object A” indicates that object A is a component of class B. That is, class A is used as a building block in the construction of class B.
- USES-A relationship: This is a using relationship which indicates that a member function of one class accepts, and therefore uses, an object of some other class as a parameter. For example, stating that “class B USES-A class A” indicates that class B objects use the facilities offered by class A objects, not that class A is used as a building block in the construction of class B.

5.5　Visual Programming

Essentially, visual programming takes OOP to the next level. The goal of visual programming is to make programming easier for programmers and more accessible to nonprogrammers by borrowing OOP concepts but exercising them in a graphical or visual way. Visual programming enables users to think more about problem solving than about handling the programming language.

Visual programming is a method of creating programs by using icons that represent common programming routines. The programmer makes connections between objects by drawing, pointing, and clicking on diagrams and icons and by interacting with flowcharts. This type of programming became necessary to develop GUI-based applications, for which earlier text-based languages are not efficient.

Visual Basic is an object-oriented programming language that was developed by Microsoft as a tool by which users of Microsoft’s Windows operating system could develop their own GUI applications. Actually, Visual Basic is more than a language — it is an entire software development package that allows a programmer to construct a GUI from predefined components (such as buttons, check boxes, text boxes, scroll bars, etc.) and to customize these components by describing how they should react to various events. In the case of a button, for example, the programmer would describe what should happen when that button is clicked.

The popularity of the Windows operating system combined with the convenience of the Visual Basic development package has promoted Visual Basic to one of the most widely used programming languages today.

5.6 Internet Programming

Many of the thousands of Internet data and information sites around the world are text-based only; that is, the user sees no graphics, animation, or video and hears no sound. The World Wide Web, however, permits all of this.

One way to build such multimedia sites on the Web is to use some relatively recently developed markup languages and programming languages: HTML, XML, Java, ActiveX, etc.

1. HTML

HTML (hyper text markup language) is an authoring language for creating Web documents. It is a type of standard for embedding codes within standard ASCII text documents to provide an integrated, two-dimensional display of text and graphics. In other words, a document created with any word processor and stored in an ASCII format can become a Web page with the addition of HTML codes.

One of the main features of HTML is its ability to insert hyperlinks into a document. Hyperlinks enable you to display another Web document simply by clicking on a link area (usually underlined or highlighted) on be current screen. One document may contain links to many other related documents. The related documents may be on the same server as the first document, or they may be on a computer halfway around the world. A link may be a word, a group of words or a picture.

Most commercial applications software packages, such as Microsoft Word, can save documents in HTML format. In addition, if you don't want to learn everything about HTML, various HTML editors and filters-commercial HTML packages will help you to create your own Web pages by choosing menu options and filling out templates; examples are Adobe's PageMill and Microsoft's FrontPage Editor. Netscape and Microsoft Internet Explorer browsers also allow users to easily create their own Web pages.

2. XML

XML (extensible markup language) makes it easy for machines to read Web sites by enabling Web developers to add descriptive "tags" to a Web page. At present, when you use your browser to find a Web site, search engines can turn up too much, so that it's difficult to pinpoint the specific site you want. XML makes Web sites smart enough to tell other machines whether they're looking at a recipe, an airline ticket, or a book for sale. XML lets Web site developers put tags on their Web pages that describe information in, for example a food recipe as "ingredients", "calories", "cooking time" and "number of portions".

3. Java

Java is a major departure from the HTML coding that makes up most Web pages. Sitting atop markup languages such as HTML and XML, Java is an object-oriented, network-friendly high-level programming language that allows programmers to build applications that can run on almost any operating system. With Java, big applications programs can be broken into

mini-applications, or “applets”, that can be downloaded off the Internet and run on any computer. Moreover, Java enables a Web page to deliver, along with visual content, applets when downloaded can make Web pages interactive.

Some microcomputers include special Java microprocessors designed to run Java software directly. However, Java is not compatible with many existing microprocessors, such as those from Intel and Motorola. For this reason, these users need to use a small “interpreter” program, called a Java Virtual Machine, that translates a Java program into a language that any computer or operating system can understand. They also need a Java-capable browser in order to view Java special effects on the Web.

Java development programs are available for programmers. In addition, Java software packages — such as Action Line, Activator Pro, Applet Ace, and Mojo — give non-programmers the ability to add multimedia effects to their Web pages, by producing applets that any Java-equipped browser can view.

Such packages can be used by anyone who understands multimedia file formats and is willing to experiment with menu options.

4. ActiveX

ActiveX was developed by Microsoft as an alternative to Java for creating interactivity on Web pages. Indeed, ActiveX is one of the two major contenders in the Web-applet war for transforming the Web into a complete interactive environment.

ActiveX is a set of controls, or reusable components that enables programs or content of almost any type to be embedded within a Web page. Whereas a Java must be downloaded each time you visit a Web site, with ActiveX, the component is downloaded only once, then stored on your hard disk for later and repeated use.

Thus, the chief characteristic of ActiveX is that it features reusable components — small modules of software code that perform specific tasks (such as spelling checker), which may be plugged seamlessly into other applications. With ActiveX you can obtain from your hard disk any file that is suitable for the Web — such as a Java applet, animation, or pop-up menu — and insert it directly into an HTML document.

Programmers can create ActiveX controls or components in a variety of programming languages, including C, C++, Visual BASIC, and Java. Thousands of ready-made ActiveX components are commercially available from software development companies.

Technical Notes to the Text

1. program，程序。程序是对计算任务的处理对象和处理规则的描述。

2. programming language，程序设计语言。Programming(程序设计)，是指设计、编制和调试程序的方法和过程。

3. software engineering，软件工程。软件工程是按照工程化的原理组织软件系统的开发。

4. flowchart，流程图。流程图指指示计算机操作步骤、过程的一组符号、方向标志符和其他表示来确定操作序列(按画出的流程图)。流程图还能使设计者对所需的过程概念清晰，对程序的每一步和每一条目有直观印象。

5. coding，编码。编码是指使用计算机语言来表达对计算机的要求。

6. documentation，文档。一种辅助工具,如流程图、文本材料、用户手册,用来理解信息系统及其各部分的结构和预期的用途。

7. machine language，机器语言。其优点是能被计算机直接理解和执行，无须翻译，效率高，执行速度快；缺点是不直观、容易出错、面向机器、通用性差。

8. assembly language，汇编语言。汇编语言指符号化的机器语言(符号语言)，用助记符的符号来代替机器语言中的操作码，用地址符号来代替机器语言中的地址码，也是一种面向机器的语言。其优点是在程序可读性方面、维护性方面比机器语言有所加强，同时也保持了机器语言执行速度快、占用存储空间小等优点；缺点是面向机器缺乏通用性、易出错、需要由汇编程序将其翻译成机器语言等。

9. high-level language，高级语言，即第 3 代语言——面向过程的语言。编程时必须告诉计算机“如何做”，即算法。其与人类的自然语言及数学语言比较接近，通用性强，即与机器的硬件无关。典型的语言有 BASIC、FORTRAN、PASCAL、C、COBOL 等。

10. artificial intelligence (AI)，人工智能。智能模拟机器用以执行通常认为要具备智力才能完成活动的能力。

11. compile，编译。通过编译程序，将高级语言源程序一次性全部“翻译”成机器语言表达的目标程序后，再执行目标程序。compiler，编译程序，是将高级语言编写的程序翻译成等价的机器语言程序或汇编语言程序的处理系统。

12. assemble，汇编。将汇编语言书写的源程序翻译成等价的机器语言程序的过程。

13. assembler，汇编程序。将汇编语言书写的源程序翻译成等价的机器语言程序的处理系统。

14. source code，源代码，源程序。其指被翻译的汇编语言程序或高级语言程序。

15. object code，目标代码。翻译后的机器语言程序。

16. OOP (object oriented programming)，面向对象程序设计。面向对象程序设计是一种运用对象、类、继承、封装、聚合、消息传递、多态性等概念来构造软件系统的程序设计方法。

17. object，对象。对象是问题域或实现域中某些事物的一个抽象，它反映该事物在系统中需要保存的信息和发挥的作用；它是一组属性和有权对这些属性进行操作的一组服务的封装体。封装(encapsulation)就是把对象的属性和服务结合成一个独立的系统单位，并尽可能隐蔽对象的内部细节。

18. class，类。类是具有相同属性和服务的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和服务两个主要部分。类有时也被称为对象类(object class)。

19. inheritance，继承。在面向对象方法中，子类无条件拥有超类所有特征的机制称为继承。继承性是对具有层次关系的类的属性和操作进行共享的一种方式。

20. overload，重载。重载是指在子类中对继承来的属性或服务进行重新定义。

21. message，消息。消息指一个对象调用另一个对象的方法(行为)所传递的一种请求信息。

22. binding，绑定。一个对象(或事物)与其某种属性建立某种联系的过程。例如，一个变

量与其类型或值建立联系，一个进程与一个处理器建立联系等。这种联系的建立，实际上就是建立了某种“约束”。在程序设计语言中，绑定是指把数据名转换为机器地址的过程，把类型或值赋予变量或参数的过程。绑定分为静态绑定和动态绑定。

23. polymorphism，多态性。多态意味着“多种形式”。在面向对象技术中，多态是指一种行为可以被不同的对象/类以不同的方式执行。

24. visual programming，可视化编程。一种用图形符号描述计算任务的处理对象和处理过程的编程方法。

25. HTML (hyper text markup language)，超文本标记语言。超文本(hyper text)采用了非线性的网状结构，使用户更快、更精确地找到需要的信息。

26. debug，调试，排错。其指检测、跟踪并排除计算机程序或其他软件中的错误。

27. 4GL (the fourth generation language)，第 4 代语言。其指一种便于非计算机专业人员掌握使用的程序设计语言，具有面向问题、非过程化、接近于自然语言、可视化等特点。典型的 4GL 有数据库查询语言、报表生成器、应用生成器、电子表格等。

Word Bank to the Text

A. Useful new words

influence	*v*. 影响，改变
module	*n*. 模数，模块
maintenance	*n*. 维护，保持
abstraction	*n*. 提取
exception	*n*. 除外，例外
procedural	*adj*. 程序上的
alternative	*adj*. 选择性的，二中择一的
emulate	*v*. 仿效
compile	*v*. 编译，编辑
perspective	*n*. 观点，看法
executable	*adj*. 可执行的，可实行的
illustrate	*v*. 举例说明，图解
oriented	*adj*. 以……为方向[目的]的，面向的
encapsulate	*v*. 装入胶囊，压缩
implement	*v*. 贯彻，实现
inheritance	*n*. 继承
restate	*v*. 重新叙述，重申
derive	*v*. 得来，得到
corresponding	*adj*. 相应的，对应的
successive	*adj*. 继承的，连续的
refinement	*n*. 精致，精巧

reusability	*n*. 可重用性
entity	*n*. 实体
graphical	*adj*. 绘成图画似的，绘画的
customize	*v*. [计]定制，用户化
embed	*v*. 使插入，使嵌入
integrate	*v*. 使成整体，使一体化
extensible	*adj*. 可展开的，可扩张的
transform	*v*. 转换，改变

B. Useful expressions

be familiar with	熟悉
have knowledge of	了解
be involved with	参与
be classified into	分类
up to date	最新的
and so on	等等
be compatible with	相兼容
set forth	阐明，提出
as mentioned above	如上所述
derive from	得自，由来
evolve from	由……进化
perform the tasks	执行任务
be depicted as	被描述为
a distinction between…and…	……和……之间的差别
be defined as	被定义为
have an influence on	影响
most importantly	更重要的是
be concerned with	牵涉到，与……有关
be responsible for	对……负责
serve the purpose of	服务于……的目的
meet the specifications	符合规范
deal with	处理
inherit from	继承
be accessible to	可以到达的，容易取得的
interact with	互动
combine with	与……结合
fill out	填写
at present	现在，目前
be willing to	乐于

transform into	转变为
be embedded within	嵌入
plug into	插入，接通

C. Technical terms and proper names

program	程序
programming language	程序设计语言
software engineering	软件工程
pseudocode	伪码
flowchart	流程图
coding	编码
program testing	程序测试
desk-checking	手工检查
documentation	文档
user documentation	用户文档
operator documentation	操作员文档
programmer documentation	程序员文档
machine language	机器语言
assembly language	汇编语言
high-level language	高级语言
very-high-level language	超高级语言
RAD (rapid application development)	快速应用开发
natural language	自然语言
artificial intelligence (AI)	人工智能
compile	编译
assemble	汇编
source code	源代码
object code	目标代码
linker	连接器
executable file	可执行文件
object-oriented programming	面向对象的程序设计
object	对象
class	类
ADT (abstract data type)	抽象数据类型
member variable	成员变量
class variable	类变量
member function	成员函数
inheritance	继承
derived class	派生类

overload	超载
message	消息
static binding	静态绑定
dynamic binding	动态绑定
polymorphism	多态性
visual programming	可视化编程
markup language	标记语言
HTML (hyper text markup language)	超文本标记语言
hyperlink	超链接
XML (extensible markup language)	可扩展标记语言
Java virtual machine	Java 虚拟机

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. In programming, the documentation is classified into _______, _______ and ________.
2. A _________is a list of instructions or statements for directing the computer to perform a required data-processing task.
3. The lowest level of programming languages is _______.
4. Very-high-level languages are often called ________.
5. ________ allow questions or commands to be framed in a more conversational way or in alternative forms.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The computer hardware recognizes only assembly language instruction. ( )
2. A program written in the assembly language of one microprocessor can run on a computer that has a different microprocessor. ( )
3. Assembly languages are platform-independent, but high-level languages are not platform-independent. ( )
4. The 4GLs are also called nonprocedural languages. ( )
5. Each assembly language instruction corresponds to one unique machine code instruction. ( )

Ⅲ. Answer the following questions.

1. What are the five steps of programming?
2. How many types of programming languages do you know?

3. What are the differences between compilation process and assembly process?

Ⅳ. Translate the following into Chinese.

1. derived class	2. inheritance	3. markup language
4. hyperlink	5. Java virtual machine

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

exception	successive	alternative
customize	influence	artificial
perspective	maintenance	successful

1. This brand of tinned beans contains no ______ coloring.
2. Jackson was the winner for a second ______ year.
3. This control allows photographers to ______ the camera's basic settings.
4. The way was blocked, so we went by an ______ road.
5. He took a course to learn about car ______.
6. My teacher ______ my decision to study science.
7. There is an ______ to this grammatical rule.
8. Most literature on the subject of immigrants in France has been written from the ______ of the French themselves.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "interpret" and fill in the sentences with the right word.

Example: By allowing the objects to determine how a **message** should be **interpreted**, the responsibility of implementing system modifications and additions can be shifted away from the supplier of a class to the consumers of a class.

1. The court has been granted the authority to interpret our la______.
2. The ability to recognize and interpret figurative la______ may help us fully understand a writer's meaning.
3. Effective readers know how to recognize and interpret fi______ of speech in order to understand the author's meaning more deeply and think about ideas in new ways.
4. Am I to interpret your si______ as acceptance or refusal?
5. Look for words that interpret one's op______ such as pretty, ugly, handsome, dangerous, evil, attractive, well-dressed, good, etc.
6. He gave a jerky ge______ which was interpreted by Donald as a gesture of dismissal.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

A total computer system includes both hardware system and software system. Hardware consists of the __1__ components and all associated equipment. Software refers to the programs that are __2__ for the computer. It is possible to be familiar with various __3__ of computer software without being concerned with details of how the computer hardware operates.

A programming language is a language used to write computer programs, which involve a computer __4__ some kind of computation or algorithm and possibly control over __5__ devices such as printers, robots, and so on. Programming languages differ from __6__ languages in that natural languages are only used for interaction between people, while programming languages also allow humans to communicate __7__ to machines. Some programming languages are used by one device __8__ control another. A prominent __9__ of programming languages is to provide instructions to a computer. Thousands of different programming languages have been __10__, and new languages are created every year.

1. A. mental	B. possible	C. essential	D. physical
2. A. design	B. spoken	C. written	D. made
3. A. parts	B. kinds	C. types	D. aspects
4. A. performed	B. to perform	C. performing	D. performs
5. A. intenal	B. external	C. inside	D. outside
6. A. natural	B. Human	C. computer	D. artificial
7. A. orders	B. instructions	C. codes	D. calls
8. A. to	B. for	C. with	D. over
9. A. show	B. research	C. purpose	D. study
10. A. creating	B. created	C. to create	D. creation

Translation

Ⅷ. Translate the following into Chinese.

1. Documentation is needed for everyone who will be involved with the program—users, operators, and programmers.

2. Rather, programs written in a high-level language or assembly language are converted to machine language, which is then executed by the computer.

3. The corresponding programs set forth precise procedures, or series of instructions, and the programmer has to follow a proper order of actions to solve a problem.

4. 4GLs may not entirely replace third-generation languages because they are usually focused on specific tasks and hence offer fewer options.

5. Inheritance is the means by which objects of a class can access member variables and functions contained in a previously defined class, without having to restate those definitions.

Chapter 6
Operating System

Pre-reading Questions

1. What is an operating system?
2. What are the functions of the operating system?
3. How many operating systems do you know? Name at least four of them.

The software system can be divided into two broad categories: application software and system software. Application software consists of the program for performing tasks particular to the machine's utilization. In contrast to application software, system software comprises a large number of programs. These programs start up the computer and function as the principal coordinator of all hardware components and application software. Without system software loaded into RAM of your computer, your hardware and application software are useless.

System software can be grouped into three basic parts: operating system, utility software, and language translators. The language translators have been introduced in Chapter 5. The majority of an installation's utility software consists of programs for performing activities that are fundamental to computer installations yet not included in the operating system. In a sense, utility software consists of software units that extend the capabilities of the operating system.

The distinction between application software and utility software is often vague. From our point of view, the distinction is whether the package is part of the software infrastructure. Thus a new application may evolve into a utility if it becomes a fundamental tool. The distinction between utility software and the operating system is equally vague. Some systems implement the software for providing such basic services as listing files in mass storage as utility software; others include it within the operating system. The operating system, the most important system software component, consists of the master programs called the supervisor that control the execution of application programs and act as an interface between the user of a computer and the computer hardware.

6.1 Operating System Overview

1. Operating System Objectives and Functions

(1) The Operating System as a User/Computer Interface

The portion of an operating system that defines the interface between the operating system and its users is often called the shell. The job of the shell is to communicate with the user, or users,

of the computer. Modern shells perform this task by means of a graphical user interface (GUI) in which objects to be manipulated, such as files and programs, are represented pictorially on the monitor screen as icons. These systems allow users to issue commands by pointing to and pushing these icons on the screen by mouse. Older shells communicate via users through textual messages using a keyboard and monitor screen.

(2) The Operating System as Resource Manager

A computer is a set of resources for the movement, storage, and processing of data and for the control of these functions. An operating system's kernel is responsible for managing these resources. In contrast to an operating system's shell, the internal part of an operating system is often called its kernel. A kernel contains those software components that perform the very basic functions required by the computer installation. An operating system's kernel consists of the file management, memory management, I/O management, and the scheduler, which we study in the next section.

2. Major Achievements

Operating systems are among the most complex pieces of software yet developed. The major achievements of operating systems are as follows.

(1) Process Management

The most fundamental building block in a modern operating system is the process. The principal function of the operating system is to create, manage, and terminate processes. While processes are active, the operating system must see that each is allocated time for execution by the processor, coordinate their activities, manage conflicting demands, and allocate system resources to processes.

To perform its process-management functions, the operating system must maintain a description of each process. Each process is represented by a process image, which includes the address space within which the process executes and a process control block. The latter contains all the information that is required by the operating system to manage the process, including its current state, resources allocated to it, priority and other relevant data.

During its lifetime, a process moves among a number of states. The most important of these are Ready, Running, and Blocked. A Ready process is one that is not currently executing but that is ready to be executed as soon as the operating system dispatches it. The Running process is that process which is currently being executed by the processor. In a multiple-processor system, more than one process can be in this state. A Blocked process is waiting for the completion of some event, such as an I/O operation.

A Running process is interrupted either by an interrupt, which is an event that occurs outside the process and which is recognized by the processor, or by executing a supervisor call to the operating system. In either case, the processor performs a context switch, transferring control to an operating-system routine. After it has completed necessary work, the operating system may resume the interrupted process or switch to some other process.

(2) Mutual Exclusion

The central themes of modern operating systems are multiprogramming, multiprocessing, and distributed processing. Fundamental to these themes, and fundamental to the technology of operating-system design, is concurrency. When multiple processes are executing concurrently, either actually in the case of a multiprocessor system or virtually in the case of a single-processor multiprogramming system, issues of conflict resolution and cooperation arise.

Concurrent processes may interact in a number of ways. Processes that are unaware of each other may nevertheless compete for resources, such as processor time and access to I/O devices. Processes may be indirectly aware of one another because they share access to a common object, such as a block of main memory or a file. And processes may be directly aware of each other and cooperate by the exchange of information. The key issues that arise in these interactions are mutual exclusion and deadlock.

Mutual exclusion is a condition in which there is a set of concurrent processes, only one of which is able to access a given resource or perform a given function at any time. Mutual-exclusion techniques can be used to resolve conflicts, such as competition for resources, and to synchronize processes so that they can cooperate. An example of the latter is the producer/consumer model, in which one process is putting data into a buffer, and one or more processes are extracting data from that buffer.

A number of software algorithms for providing mutual exclusion have been developed, of which the best known is Dekker's algorithm. The software approach is likely to have high processing overhead, and the risk of logical errors is high. A second approach to supporting mutual exclusion involves the use of special purpose machine instructions. This approach reduces overhead, but it is still inefficient because it uses busy-waiting.

Another approach to supporting mutual exclusion is to provide features within the operating system. Two of the most common techniques are semaphores and message facilities. Semaphores are used for signaling among processes and can be used readily to enforce a mutual-exclusion discipline. Messages are useful for the enforcement of mutual exclusion and also provide an effective means of interprocess communication.

(3) Deadlock

Deadlock is the blocking of a set of processes that either compete for system resources or communicate with each other. The blockage is permanent unless the operating system takes some extraordinary action, such as killing one or more processes or forcing one or more processes to backtrack. Deadlock may involve reusable resources or consumable resources. A consumable resource is one that is destroyed when it is acquired by a process; examples include messages and information in I/O buffers. A reusable resource is one that is not depleted or destroyed by use, such as an I/O channel or a region of memory.

There are three general approaches to deal with deadlock: prevention, detection, and avoidance. Deadlock prevention guarantees that deadlock will not occur by assuring that one of the necessary conditions for deadlock is not met. Deadlock detection is needed if the operating system

is always willing to grant resource requests; periodically, the operating system must check for deadlock and take action to break the deadlock. Deadlock avoidance involves the analysis of each new resource request to determine if it could lead to deadlock, and granting it only if deadlock is not possible.

(4) Memory Management

One of the most important and most complex tasks of an operating system is memory management. Memory management involves treating main memory as a resource to be allocated to and shared among a number of active processes. To efficiently use the processor and the I/O facilities, it is desirable to maintain as many processes in main memory as possible. In addition, it is desirable to free programmers from size restriction in program development.

The way to address both of these concerns is to use virtual memory. With virtual memory, all address references are logical references that are translated at run time to real addresses. This use allows a process to be located anywhere in main memory and for that location to change over time. Virtual memory also allows a process to be broken up into pieces. These pieces need not be contiguously located in main memory during execution, and indeed it is not even necessary for all the pieces of the process to be in main memory during execution.

Two basic approaches to providing virtual memory are paging and segmentation. With paging, each process is divided into relatively small, fixed-size pages. Segmentation provides for the use of pieces of varying sizes. It is also possible to combine segmentation and paging in a single memory-management scheme.

A virtual-memory-management scheme requires both hardware and software support. The hardware support is provided by the processor. The support includes dynamic translation of virtual addresses to physical addresses and the generation of an interrupt when a referenced page or segment is not in main memory. Such an interrupt triggers the memory-management software in the operating system.

A number of design issues relate to operating system support for memory management as follows:

- Fetch policy: Process pages can be brought in on demand, or a prepaging policy can be used; the latter clusters the input activity by bringing in a number of pages at once.
- Placement policy: With a pure segmentation system, an incoming segment must be fit into an available space in memory.
- Replacement policy: When memory is full, a decision must be made as to which page or pages are to be replaced.
- Resident set management: The operating system must decide how much main memory to allocate to a particular process when that process is swapped in. This can be a static allocation made at process creation time, or it can change dynamically.
- Cleaning policy: Modified process pages can be written out at the time of replacement, or a precleaning policy can be used; the latter clusters the output activity by writing out a number of pages at once.

- Load control: Load control is concerned with determining the number of processes that will be resident in main memory at any given time.

(5) Uniprocessor Scheduling

The operating system must make three types of scheduling decisions with respect to the execution of processes. Long-term scheduling determines when new processes are admitted to the system. Medium-term scheduling is part of the swapping function and determines when a program is brought partially or fully into main memory so that it may be executed. Short-term scheduling determines which ready process will be executed next by the processor. This section focuses on the issues relating to short-term scheduling.

A variety of criteria are used in designing the short-term scheduler. Some of these criteria relate to the behavior of the system as perceived by the individual user (user oriented), whereas others view the total effectiveness of the system in meeting the needs of all users (system oriented). Some of the criteria relate specifically to some quantitative measure of performance, whereas others are more qualitative in nature. From a user’s point of view, response time is generally the most important characteristic of a system, whereas from a system point of view, throughput or processor utilization is important.

A variety of algorithms have been developed for making the short-term scheduling decision among all ready processes. These include the following:

- First-come, first-served: Select the process that has been waiting the longest for service.
- Round-robin: Use time-slicing to limit any running process to a short burst of processor time, and rotate among all ready processes.
- Shortest process next: Select the process with the shortest expected process time and do not preempt the process.
- Shortest remaining time: Select the process with the shortest expected remaining process time. A process may be preempted when another process becomes ready.
- Highest response ratio next: Base the scheduling decision on an estimate of normalized turnaround time.
- Feedback: Establish a set of scheduling queues and allocate processes to queues based on execution history and other criteria.

The choice of scheduling algorithm will depend on expected performance and on implementation complexity.

(6) Multiprocessor and Real-time Scheduling

With a tightly coupled multiprocessor, multiple processors have access to the same main memory. In this configuration, the scheduling structure is somewhat more complex. For example, a given process may be assigned to the same processor for its entire life, or it may be dispatched to any processor each time when it enters the running state. Performance studies suggest that the differences among various scheduling algorithms are less significant in a multiprocessor system.

A real-time process or task is one that is executed in connection with some process or function or set of events external to the computer system and that must meet one or more deadlines

to interact effectively and correctly with the external environment. A real-time operating system is one that must manage real-time processes. In this context, the traditional criteria for choosing a scheduling algorithm do not apply. Rather, the key factor is the meeting of deadlines. Algorithms that rely heavily on preemption and on reacting to relative deadlines are appropriate in this context.

(7) I/O Management

The computer system's interface to the outside world is its I/O architecture. This architecture is designed to provide a systematic means of controlling interaction with the outside world and to provide the operating system with the information it needs to manage I/O activity effectively.

The I/O function is generally broken up into a number of layers, with lower layers dealing with details that are closer to the physical functions to be performed, and higher layers dealing with I/O in a logical and generic fashion. The result is that changes in hardware parameters need not affect most of the I/O software.

A key aspect of I/O is the use of buffers that are controlled by I/O utilities rather than by application processes. Buffering smooth out the differences between the internal speeds of the computer system and the speeds of I/O devices. The use of buffers also decouples the actual I/O transfer from the address space of the application process, which allows the operating system more flexibility in performing its memory-management function.

The aspect of I/O that has the greatest impact on overall system performance is disk I/O. Accordingly, there has been greater research and design effort in this area than in any other kind of I/O. Two of the most widely used approaches to improving disk I/O performance are disk scheduling and the disk cache.

At any time, there may be a queue of requests for I/O on the same disk. It is the object of disk scheduling to satisfy these requests in a way that minimizes the mechanical seek time of the disk and hence improves performance. The physical layout of pending requests plus considerations of locality come into play.

A disk cache is a buffer, usually kept in main memory that functions as a cache of disk blocks between disk memory and the rest of main memory. Because of the principle of locality, the use of a disk cache should substantially reduce the number of block I/O transfers between main memory and disk.

(8) File Management

A file-management system is a set of system software that provides services to users and applications in the use of files, including file access, directory maintenance, and access control. The file-management system is typically viewed as a system service that itself is served by the operating system, rather than being part of the operating system itself. However, in any system, at least part of the file-management function is performed by the operating system.

A file consists of a collection of records. The way in which these records may be accessed determines its logical organization and to some extent its physical organization on disk. If a file is primarily to be processed as a whole, then a sequential file organization is the simplest and most appropriate. If sequential access is needed but random access to an individual file is also desired,

then an indexed sequential file may give the best performance. If access to the file is principally at random, then an indexed file or hashed file may be the most appropriate.

Whatever file structure is chosen, a directory service is also needed to allow files to be organized in a hierarchical fashion. This organization is useful to the user in keeping track of files and is useful to the file-management system in providing access control and other services to users.

File records, even when of fixed size, generally do not conform to the size of a physical disk block. Accordingly, some sort of blocking strategy is needed. A trade-off among complexity performance, and use of space determines the blocking strategy to be used.

6.2 Operating System Platform

The type of processor used in a computer determines the type of machine language it uses. And the computer's operating system is created to work with that particular type of machine language. Thus, the processor model and the operating system determine the platform — that is, the type of computer architecture, or family. The PC and the Apple Macintosh are two common platforms. For the most part, software created for one type of platform will not run, without special arrangements, on other platforms.

Common operating systems used on mainframes and midsize computers are MVS, VM, OS/390, and VAX/VMS. In this section, we describe the following microcomputer operating systems.

1. DOS and Windows 3.x

Command-driven DOS (Disk Operating System) was very popular in the 1980s and early 1990s. IBM's version of DOS is called PC-DOS and Microsoft's version is called MS-DOS. Except for subtle differences, PC-DOS and MS-DOS are identical. The initial version, DOS 1.0, was released in August 1981. It consisted of 4,000 lines of assembly-language source code and ran in 8KB of memory by using the Intel 8086 microprocessor.

When IBM developed a hard-disk-based personal computer, the PC XT, Microsoft developed DOS 2.0, released in 1983. It contained support for the hard disk and provided for hierarchical directories. The new release allowed directories to contain subdirectories as well as files. The new release also contained a richer set of commands embedded in the operating system to provide functions that had to be performed by external programs provided as utilities with release 1. Among the capabilities added were several UNIX-like features, such as I/O redirection, which is the ability to change the input or output identity for a given application, and background printing. The memory-resident portion grew to 24KB.

When IBM announced the PC AT in 1984, Microsoft introduced DOS 3.0. The AT contained the Intel 80286 processor, which provided extended addressing and memory-protection features. DOS 3.1, released in 1984, contained support for networking of PCs. DOS 3.3, released in 1987,

provided support for the new line of IBM machines, the PS/2. DOS 7.0 was issued in 1996 and includes enhanced support for managing networks and the latest microprocessors.

By this time, DOS was being used in an environment far beyond its capabilities. The introduction of the 80486 and then the Intel Pentium chip provided power and features that simply could not be exploited by the simple-minded DOS. Meanwhile, beginning in the early 1980s, Microsoft has begun development of a graphical user interface that would be interposed between the user and DOS.

In 1985, Microsoft released Windows 1.0, an operating environment that lays a graphical user interface shell around DOS and extends DOS's capabilities. Version 2.0 was released in 1988, but it wasn't until version 3.0 was released in 1990 that Windows really took off and created an industry tied to its GUI, multitasking capabilities, and ability to manage large amounts of memory. There have been several releases of Windows 3 — 3.0, 3.1, and 3.11. With Windows 3.x, you display your work in one or more windows on the desktop. You can easily switch and move data among windows.

2. Windows 9x

Windows 95, released in 1995, was the major upgrade designed to replace DOS and Windows 3.x. A true multitasking operating system, Windows 95 does not require the separate MS-DOS program. The GUI is not the shell; instead it is integrated into the operating system. Like Windows 3.x, Windows 95 uses windows and a desktop. Among the many features included in Windows 95 are support for longer filenames, e-mail, fax transmission, multimedia, and Plug and Play, which simplifies the process of installing new hardware.

Most important, though, is the fact that Windows 95 is a 32-bit operating system, meaning that it can work with 32 bits of data at one time. This is twice as much as can be processed by any version of DOS, which can only work with data in 16-bit chunks. What this means to you is that applications written for Windows 95 run faster and you can work with multiple applications at the same time without a noticeable delay. Also, with Windows 95, you are able to realize the potential and power of today's more sophisticated microprocessors.

Windows 98, which models its interface after Windows 95, includes several enhanced capabilities over Windows 95 — primarily, its support of the Internet. With Windows 98, you can open documents stored locally on your computer, documents stored on your company's network, and documents stored on the Web without having to open a separate Web-browser window. In addition, Windows 98 provides support for huge hard disks, state-of-the-art hardware such as DVD disks and the Universal Serial Bus standards, and additional commands for customizing the user interface.

A simplified version of Windows 9x, Windows CE, is used for palmtop computers.

3. Windows NT/Windows 2000

Whereas Windows 9x is basically consumer-oriented, Windows NT, for New Technology, is

business-oriented. Providing a similar interface to Windows 9x, it is a multitasking, multiprocessing operating system with built-in support for large networks of computers — that is, Windows NT is a multi-user system. Multi-user platforms support workgroup computing, situations in which LANs are set up to allow users to share files, databases, and applications.

Windows 2000 combines the Windows NT line with Windows 9x.

4. OS/2 Warp

OS/2 was initially released in April 1987 as IBM's contender for the next mainstream operating system. OS/2 (Operating System 2) was designed to run on IBM and IBM-compatible microcomputers. Unfortunately, because of an array of management and marketing disasters, IBM slipped far behind Microsoft in developing an installed base for OS/2. In late 1994 IBM unveiled a souped-up version of OS/2, called OS/2 Warp. OS/2 Warp is similar to Windows NT. IBM has even started shipping Windows NT on some of its lower-end systems. Although the future of OS/2 Warp is uncertain, IBM continues to support its approximately 10 million Warp users. The latest versions of Warp are available online and can be downloaded from IBM's Web site.

5. UNIX

UNIX was invented more than two decades ago by American Telephone & Telegraph (AT & T), making it the oldest operating system still used today. UNIX is a multi-user, multitasking operating system with built-in networking capability. Because it can run with relatively simple modifications on most types of computers — from micros to minis to mainframes — UNIX is called a portable operating system.

These early versions of UNIX were quite popular within Bell Labs. In 1974, the UNIX system was described in a technical journal for the first time. This spurred great interest in the system. Licenses for UNIX were provided to commercial institutions as well as universities. The first widely available version outside Bell Labs was version 6, in 1976. The follow-on version 7, released in 1978, is the ancestor of most modern UNIX systems. The most important of the non-AT & T systems to be developed was done at the University of California at Berkeley. It was called UNIX BSD and ran first on PDP and then VAX machines. AT&T continued to develop and refine the system. By 1982, Bell Labs had combined several AT&T variants of UNIX into a single system that was marketed commercially as UNIX System III. A number of features were later added to the operating system to produce UNIX System V.

6. Linux

Another PC-compatible operating system option is Linux, a version of UNIX. Linus Torvalds developed Linux in 1990 while he was a computer science student at Helsinki University in Finland. He developed it as freeware offering UNIX's power without the hefty price. Free Linux versions can be downloaded from hundreds of Internet sites.

While Linux has ample power to run many Windows-based applications, few software developers currently design their software to be used specifically with Linux. However, some

software manufacturers are supporting Linux. Corel has released a Linux version of WordPerfect 8, Netscape offers a Linux-compatible version of its browser Navigator 4.04, and Caldera offers a Linux office suite, Star Office. Indeed, Red Hat Software and other commercial Linux distributors have simplified installation.

Some hardware manufacturers are also supporting Linux. Compaq, Dell, and Hewlett-Packard are shipping some server computers with Linux instead of Windows NT.

Technical Notes to the Text

1. application software，应用软件。应用软件指为利用计算机解决各种实际问题而编制的程序。

2. system software，系统软件。系统软件指计算机的管理、控制、维护、使用及程序安装等与硬件配套管理的基础软件。

3. operating system，操作系统。负责管理和控制计算机系统的各种资源，合理地组织计算机的工作流程，以充分发挥计算机系统的效率。其是用户和计算机之间的接口。

4. shell，操作系统的外壳程序。其是指用户与操作系统间的一种软件接口。

5. GUI (graphical user interface)，图形用户界面。在 20 世纪 80 年代计算机图形技术基础上发展起来的图形用户界面，是当今计算机用户界面的主流，它采用 WIMP 技术，即窗口(window)、图标(icon)、菜单(menu)及指点设备(pointing device)技术，采用多窗口系统为主要软件，以直接操纵为主要使用方法。

6. kernel，内核。内核是指在计算机软件中，当装入操作系统任何部分时，事先必须已经并总是存在于内存(主存)之中的操作系统中的那部分内容，主要包括一些经过严格测试而准确无误的例行程序，它们可以实现基本的装入和管理功能。

7. batch processing，批处理。批处理是指以提交作业的方式把任务的执行过程集中起来，经过合理搭配，通过输入装置提交给计算机系统。

8. multitasking，多任务。在多道程序设计中，计算机内存中同时存放几道相互独立的程序，它们在管理程序的控制下，相互穿插地运行。

9. time-sharing system，分时系统。分时系统一般运行在一台主机带有若干台终端的系统，即若干台终端共享一台 CPU 设备(即联机操作方式)。它采用分时技术，用时间轮流转的办法使一台计算机能够为多个终端用户服务，执行每个用户提出的任务。UNIX 是一个典型的分时操作系统。分时(time-sharing)是指操作系统按照一定的分配原则把 CPU 的整个机时划分为若干个极小的时间片，然后将其依次轮流地分配给各终端用户使用。

10. process management，进程管理。进程管理负责进程的创建、调度、执行和撤销。

11. distributed processing，分布式处理。将不同地点或具有不同功能的或拥有不同数据的多台计算机用通信网络连接起来，在控制系统的统一管理控制下，协调地完成信息处理任务。

12. deadlock，死锁。死锁是指多个进程循环等待他方占有的资源而无限地僵持下去的局面。

13. I/O buffer，输入/输出缓冲区。其是指一种无须主程序干预就可以将数据字传输给内

存或从内存调出的缓冲器。可通过编程使输入输出传输完成时计算机产生一个内部中断。

14. I/O channel，输入/输出通道。允许在内存储器和输入输出设备之间独立通信的一种设备，它控制任何的外部设备，并实现信息传输的所有正确性校验。

15. paging，分页。分页指主存与辅存之间的页面传送过程。

16. real-time process，实时处理。实时处理指计算机可以随时对发生的外部事件做出及时的响应，在严格的时间内完成相应处理。

17. file management，文件管理。文件管理负责向用户提供创建文件、撤销文件、读写文件、打开和关闭文件等功能。

Word Bank to the Text

A. Useful new words

embark	*v*. 着手，从事
principal	*adj*. 主要的，最重要的
kernel	*n*. 核心，内核
allot	*v*. (按份额)分配，分派
dismount	*v*. 取下，卸下
submit	*v*. 提交，递交
dedicated	*adj*. 专用的
batch	*n*. 一批，一组，大量
terminal	*n*. 终点站，终端
interfere	*v*. 干涉，干预
authorize	*v*. 批准
coordinate	*v*. 协调，调整
priority	*n*. 优先，优先权
mutual	*adj*. 共同的
feature	*n*. 特征，特色
enforce	*v*. 推行，实施
deadlock	*n*. 僵局，僵持
backtrack	*v*. (由原路)退回
segmentation	*n*. 分割
trigger	*n*. 引发，触发
criteria	*n*. 标准
preemption	*n*. 先买权，先买，先占
parameter	*n*. 参数
flexibility	*n*. 弹性，适应性，机动性

locality	*n*. 位置，地点
identical	*adj*. 同一的，同样的
initial	*adj*. 开始的，最初的
exploit	*v*. 开拓，开发
interpose	*v*. 置于……之间，使介入
enhance	*v*. 提高，增强
approximately	*adv*. 近似地，大约

B. Useful expressions

in contrast	相反，大不相同
group into	分组，分类
from one's point of view	从某人的观点来看
evolve into	发展成，进化成
act as	担当
interact with	相互作用，相互影响
sign up for	签约承担义务
result in	导致
run into	偶遇，陷入
have access to	有权进入
be known as	被认为是，称为
rely on	依靠
carry out	执行
just as	正如
a host of	许多，一大群
interfere with	干涉，干扰
allocate time for	为……分配时间
be aware of	意识到
deal with	处理，应对
fit into	适合
in connection with	与……有关，连同
break up into	分解成，分割成
to some extent	在某种程度上
for the most part	在极大程度上

C. Technical terms and proper names

application software	应用软件
system software	系统软件
utility software	实用软件
operating system (OS)	操作系统

shell	操作系统的外壳程序
graphical user interface (GUI)	图形用户界面
kernel	内核
serial processing	串行处理
job	作业
batch processing	批处理
simple batch system	简单批处理系统
multiprogrammed batch system	多道程序批处理系统
monitor	监控程序
scheduler	调度程序
multiprogramming	多道程序
multitasking	多任务
time-sharing system	分时系统
uniprogramming	单道程序
process	进程
process management	进程管理
process control block	进程控制块
mutual exclusion	互斥
multiprocessing	多处理，多进程
distributed processing	分布式处理
concurrent process	并发处理
deadlock	死锁
synchronize process	同步处理
semaphore	信号量
reusable resource	可复用性资源
I/O buffer	输入/输出缓冲区
I/O channel	输入/输出通道
deadlock prevention	死锁预防
deadlock detection	死锁监测
deadlock avoidance	死锁避免
virtual memory	虚拟内存
logical reference	逻辑引用
real address	实地址
paging	分页
segmentation	分段
virtual address	虚拟地址
physical address	物理地址
real-time process	实时处理

file management 文件管理
plug and play (PnP) 即插即用

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. The portion of an operating system that defines the interface between the operating system and its users is often called the _______.
2. An operating system's kernel consists of the file management, _______, I/O management, and the scheduler.
3. _______ contains all the information that is required by the operating system to manage the process, including its current state, resources allocated to it, priority and other relevant data.
4. A _____ process is waiting for the completion of some event, such as an I/O operation.
5. There are three general approaches to dealing with deadlock: _______, _______, and_______.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. A blocked process is one that is not currently executing but one that is ready to be executed as soon as the operating system dispatches it. ( )
2. Semaphores are used for signaling among processes and can be used readily to enforce a mutual-exclusion discipline. ( )
3. A consumable resource is one that is not depleted or destroyed by use, such as an I/O channel or a region of memory. ( )
4. Windows 3.x are support for longer filenames and Plug and Play. ( )
5. UNIX is a multi-user, multitasking operating system with built-in networking capability. ( )

Ⅲ. Answer the following questions.

1. How many parts does the system software have?
2. What are the objectives and functions of the operating system?
3. What is a deadlock? Are there general approaches to deal with deadlock?
4. What are the major achievements of the operating system?

Ⅳ. Translate the following into Chinese.

1. graphical user interface (GUI) 2. process control block 3. mutual exclusion
4. semaphore 5. concurrent process

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

criteria	authorize	comprise
initial	priority	submit
flexibility	mutual	principal

1. The special cabinet committee _______ Mr. Brown, Mr. Mandelson, and Mr. Straw.
2. Once the _______ contradiction is grasped, all problems will be readily solved.
3. I hope you can _______ you term papers before the deadline.
4. The East and the West can work together for their ________ benefit and progress.
5. Getting your _______ in order is a good way not to waste energy on meaningless pursuits.
6. We are willing to ________ the president to use force if necessary.
7. The ________ of the lens decreases with age; it is therefore common for our sight to worsen as we get older.
8. The ________ talks were the base of the later agreement.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "integrate"and fill in the sentences with the right word.

Example: The **GUI** is not the shell; instead it is **integrated** into the operating system.

1. The idea with young children is to integrate le______ with play.
2. Our brains integrate si______ from our eyes and ears with other information from the organs in our inner ear, from our muscles and joints, and from our senses of touch and pressure.
3. Our objectives are to simplify and integrate the two sy______, to mount a determined assault on poverty and dependency, and to protect our citizens from want.
4. He tried to integrate this sc______ with the undergraduate college for he believed such a move would make the graduate school more responsive to his authority.
5. It's very difficult to integrate yo______ into a society whose culture is so different from your own.
6. The in______ can be integrated into language learning and supports communication, research, reading, and writing development.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

An operating system is the software component of a computer system that is responsible __1__ the management and coordination of activities and the __2__ of the resources of the computer. The operating system (OS) acts as a host for application programs that are run on the machine. As a host, one of the purposes of an operating system is __3__ the details of the operation of

the hardware. This relieves application programs __4__ having to manage these details and makes __5__ easier to write applications. Almost all computers, including hand-held computers, desktop computers, supercomputers, and __6__ modern video game consoles, use an operating system of some type.

Operating systems offer a number of services to application programs and users. Applications access these services through application programming __7__ (APIs) or system calls. By __8__ these interfaces, the application can request a service from the operating system, pass parameters, and receive the results of the operation. Users may also __9__ with the operating system by typing commands or using a graphical user interface (GUI).

Common contemporary operating systems include Windows XP, Mac OS X, Amiga OS, Linux and Solaris. Microsoft Windows has a significant majority of market __10__ in the desktop and notebook computer markets.

1. A. in	B. with	C. over	D. for
2. A. sharing	B. using	C. commanding	D. controlling
3. A. handling	B. to handle	C. handle	D. handled
4. A. to	B. from	C. since	D. due
5. A. that	B. which	C. this	D. it
6. A. more	B. even	C. great	D. so
7. A. internet	B. indications	C. interfaces	D. interchange
8. A. invoke	B. to invoke	C. invoking	D. invoked
9. A. interacted	B. interact	C. interactive	D. interaction
10. A. part	B. place	C. strategy	D. share

Translation

Ⅷ. Translate the following into Chinese.

1. The majority of an installation's utility software consists of programs for performing activities that are fundamental to computer installations yet not included in the operating system.

2. Modern shells perform this task by means of a graphical user interface (GUI) in which objects to be manipulated, such as files and programs, are represented pictorially on the monitor screen as icons.

3. The basic technique for a time-sharing system is to have multiple users simultaneously using the system through terminals, with the operating system interleaving the execution of each user program in a short burst, or quantum, of computation.

4. A real-time process or task is one that is executed in connection with some process or function or set of events external to the computer system and that must meet one or more deadlines to interact effectively and correctly with the external environment.

5. Providing a similar interface to Windows 9x, it is a multitasking, multiprocessing operating system with built-in support for large networks of computers — that is, Windows NT is a multi-user system.

Chapter 7
Applications Software

Pre-reading Questions

1. What is applications software?
2. Are there any differences between the systems software and the applications software?
3. How many types of applications software do you know? Give at least two examples.

Getting the computer and the peripherals — the hardware — is only the beginning. But it's your choice of software that will really determine the utility — or uselessness — of your new PC.

7.1 Applications Software Tools

Applications software is the software designed to help you solve problems specific to business or perform specific business tasks. Applications software then is the layer of software closest to you.

Basically, there are four categories of application software: productivity software, business and specialty software, entertainment software and education/reference software.

- Basic productivity software consists of programs found in most offices and probably on all computers, on personal computers and on larger computer systems. Most of the common applications software packages used today are productivity tools. Their purpose is simply to make users more productive when performing general tasks. Examples are word processing, spreadsheet, personal finance, presentation graphics, database management, group collaboration, desktop and personal information management, and web browsing applications.
- Business and specialty software, often called vertical market software, consists of programs developed for a specific business or industry. Whatever your occupation, you will probably find it also has specialty software tools available to it. This is true whether your career is dairy farmer, building contractor, police officer, dance choreographer, or chef. Examples are desktop-publishing software, project management software, computer-aided design (CAD) and manufacturing (CAM) software, drawing and painting software, multimedia authoring software.
- Entertainment software consists of action and adventure games that can be played at home through a television set or personal computer or in an entertainment arcade of the sort found in shopping malls.
- Education and reference software is used to educate and entertain. Because of the popularity

of videogames, many educational software companies have been blending educational content with some of the features popular in games. Computers alone won't boost academic performance, but they can have a positive effect on student achievement in all major subject areas, preschool through college. In addition to educational software, library search and reference software has become popular. For instance, there are CD-ROMs with encyclopedias, phone books, mailing lists, maps, and reproductions of famous art.

7.2 Common Features of Applications Software

Although applications software packages differ in their use of specific commands and functions, most of them have some features in common:

- Insertion point: The insertion point is the movable symbol on the display screen that shows you where you may enter data next. You can move the insertion point around using either the keystroke's directional arrow keys or by clicking with the mouse.
- Scrolling: Scrolling is the activity of moving quickly upward or downward through the text or other screen display. A standard computer screen displays only 20—22 lines of standard-size text. Of course, most documents are longer than that. Using the directional arrow keys or a mouse, you can move (scroll) through the display screen and into the text above and below it.
- Windows: A window is a rectangular section of the display screen with a title bar on top. Each window may show a different display, such as a word processing document in one and a spreadsheet in another.
- Menu bar: A menu is a list of command options, or choices. A menu bar is a row of menu options displayed across the top or the bottom of the screen.
- Pull-down menu: A pull-down menu is a list of command options, or choices, that is "pulled down" out of the menu bar. Pull-down menus can be opened by keystroke commands or by "clicking" (pressing) the mouse button while pointing to the title in the menu bar and then dragging the mouse pointer down. Some menus "pop up" from the menu bar and so are called pop-up menus.
- Help menu and screens: A help menu offers a choice of Help screens, specifically displayed explanations of how to perform various tasks, such as printing out a document. Having a set of Help screens is like having a built-in electronic instruction manual. Help features also include searchable topic indexes and online glossaries.
- Buttons: In a graphical interface, buttons are used to represent file names and the popular features and functions. Buttons are usually identified by a small graphic, called an icon. Most applications use toolbars to group related buttons.
- Toolbars: A toolbar is a row of on-screen buttons, usually appearing immediately below the menu bar, used to activate a variety of functions of the applications program. Toolbars

can often be customized and moved around on the screen.

- Dialog box: A dialog box is a box that appears on the screen. It is used to collect information from the user and to display helpful messages.
- Default values: Default values are standard settings employed by the computer. When the user does not specify particular margin widths in your page setup, the word processing program will use the manufacturer's default values.
- Macro: A macro is a feature that allows you to use a single keystroke, command, or toolbar button to automatically issue a predetermined series of commands. Thus, you can consolidate several keystrokes or menu selections into only one or two keystrokes. Although many people have no need for macros, you will find them quite useful if you need to continually repeat complicated patterns of keystrokes.
- OLE: Many applications softwares have the ability to integrate applications using OLE (object linking and embedding). This feature enables you to embed an object created using one application (such as graphics) into another application (such as word processing). Changes made to embedded object affect only the document that contains it. Objects can also be linked. In this case, changes made to the object are automatically made in all the linked documents that contain it. Thus OLE facilitates the sharing and manipulating of information. An object may be a document, worksheet, chart, picture, or even a sound recording.
- Clipboard: Many applications software programs allow you to copy an item from one document and then paste it into another document or application, or copy an item and place the copy in another part of the same document. The clipboard is the area where the copy is held before it is pasted.
- Tutorials and documentation: How are you going to learn a given software program? Most commercial packages come with tutorials. A tutorial is an instruction book or program that takes you through a prescribed series of steps to help you learn the product. Tutorials must be contrasted with documentation. Documentation is a user manual or reference manual that is a narrative and graphical description of a program. Documentation may be instructional, but features and functions are usually grouped by category for reference purpose. For example, in word processing documentation, all cut-and-paste features are grouped together so that you can easily look them up if you have forgotten how to perform them. Documentation may come in booklet form or on diskette or CD-ROM; it may also be available online from the manufacturer.

7.3 Productivity Software Tools

Let's now look at the various types and uses of productivity software for:

- Word processing
- Spreadsheet applications

- Personal finance
- Presentation graphics
- Database managers
- Group collaboration
- Desktop and personal information management
- Integrated applications
- Web browsers

7.3.1 Word Processing Software

Word processing software allows you to use computers to create, edit, store, and print documents. You can easily insert, delete, and move words, sentences, and paragraphs — without ever using an eraser. Word processing programs also offer a number of features for "dressing up" documents with variable margins, type sizes, and styles. The user can do all these manipulations on screen, in "wysiwyg" fashion, before printing out hardcopy. (wysiwyg stands for "what you see is what you get", meaning that the screen displays documents exactly as they will look when printed.)

Today, popular word processing programs are Microsoft Word for the PC, Word for the Mac, Corel WordPerfect for the PC, and WordPerfect for the Mac.

Word processing software also offers the following features:

1. Creating Documents

Creating a document means entering text, using the keyboard. As you type, word wrap automatically continues text on the next line when you reach the right margin. That is, the text "wraps around" to the next line.

2. Editing Documents

Editing is the act of making alterations in the content of your document. Some features of editing are insert and delete; undelete; search and replace; cut, copy, and paste; spelling checker; grammar checker; and thesaurus.

3. Formatting Documents

Formatting means determining the appearance of a document. There are many choices here, including types, spacing and columns, margins and justification, pages, headers and footers, etc.

It's worth noting that word processing programs (and indeed most forms of applications software) come from the manufacturer with default settings. Thus, for example, most word processing programs will automatically prepare a document single-spaced, justified, and with 1-inch right and left margins unless you alter these default settings, which is easy to do.

4. Printing Documents

Most word processing software gives you several options for printing. For example, you can print several copies of a document. You can print individual pages or a range of pages. You can even

preview a document before printing it out. Previewing (print previewing) means viewing a document on screen to see what it will look like in printed form. Whole pages are displayed in reduced size.

7.3.2 Spreadsheet Software

What is spreadsheet? Spreadsheet software takes its name from the accountant's columnar worksheet, which it imitates. A spreadsheet is a worksheet consisting of a collection of cells formed by the intersection of rows and columns. Each cell can store one piece of information: a number, word or phrase, or formula. Today the principal spreadsheet programs are Microsoft Excel and Lotus 1-2-3.

1. Principal Features

Spreadsheet software worksheets include the following features:

- Columns and rows: Column headings appear across the top ("A" is the name of the first column, "B" is the second, and so on). Row headings appear down the left side ("1" is the name of the first row, "2" the second, and so forth).
- Cells, cell addresses, and cell pointer: The place where a row and a column intersect is called a cell, and its position is called a cell address. For example, "A1" is the cell address for the top left cell, where column A and row 1 intersect. A cell pointer indicates where data is to be entered. The cell pointer can be moved around like the insertion point in a word processing program.
- Values: A number, date, or formula entered in a cell is called a value. The values are the actual numbers used in the worksheet — dollars, percentages, grade points, temperatures, or whatever.
- Formulas, functions, recalculation: Now we come to reason the electronic spreadsheet has taken officers by storm. Formulas are instructions for calculations. For example, a formula might be A1+A2+A3, meaning to add the contents of cells A1, A2, and A3.
  Functions are built-in formulas that perform common calculations. For instance, a function might sum or average a range of numbers or round off a number to two decimal places. An example of a function is SUM(A1:A3), meaning "Sum (add) all the numbers in the cells with cell addresses A1 through A3".
  After the values have been plugged into the spreadsheet, the formulas and functions can be used to calculate outcomes. What is revolutionary, however, is the ease with which the spreadsheet does recalculation. Recalculation is the process of recomputing values automatically, either as an ongoing process or as data is being entered of afterward, with the press of a key. With this simple feature, the hours of mind-numbing work required to manually rework paper spreadsheets has become a thing of the past.
  The recalculation feature has opened up whole new possibilities for decision making. As a user, you can create a plan, put in formulas and numbers, and then ask yourself, "What would happen if we change that detail?" — and immediately see the effect on the bottom

line. This is called the *what-if* function. For example, if you're buying a new car, the "what if " function lets you consider various options: Any number of things can be varied: total price ($10,000? $15,000?), down payment ($2000? $3000?), interest rate on the car loan (7%? 8%?), or number of months to pay (36? 48?). You can keep changing the "what if" possibilities until you arrive at a monthly payment figure that you're comfortable with.

Spreadsheets can be linked with other spreadsheets. The feature of dynamic linking allows data in one spreadsheet to be linked to and automatically update data in another spreadsheet. Thus, the amount of data being manipulated can be enormous.

2. Analytical Graphics: Creating Charts

Another useful feature of spreadsheet packages is the ability to create analytical graphics. Presented in spreadsheet forms, as rows and columns of numbers, financial data is not always easy to comprehend. Whether viewed on a monitor or printed out, analytical graphics, or business graphics, help make sales figures, economic trends, and the like easier to comprehend and analyze.

The principal examples of analytical graphics are bar charts, line graphs, and pie charts. Quite often these charts can be displayed or printed out so that they look three-dimensional. Spreadsheets can even be linked to more exciting graphics, such as digitized maps.

7.3.3 Presentation Graphics

Computer graphics can be highly complicated, such as those used in special effects for movies. Here we are concerned with just one kind of graphics, called presentation graphics.

Presentation graphics are part of presentation software, which uses graphics and data/ information from other software tools to communicate or make a presentation to others, such as clients or supervisors. Presentations may make use of some analytical graphics-bar, line, and pie charts — but they most often include bulleted lists. Examples of well-known presentation software packages are Microsoft PowerPoint, Aldus Persuasion, Lotus Freelance Graphics, and SPC Harvard Graphics.

Some presentation software packages provide artwork called clip art that can be electronically cut and pasted into the graphics. These programs also allow you to use electronic painting and drawing tools for creating lines, rectangles, and just about any other shape. Depending on the system's capabilities, you can add text, animated sequences, and sound.

7.3.4 Groupware

Most microcomputer software is written for people working alone. Groupware is software that is used on a network and serves a group of users working together on the same project. Groupware improves productivity by keeping you continually notifie about what your colleagues are thinking and doing, and vice versa.

Groupware is essentially of four types:

- Basic groupware: Exemplified by Lotus Notes, this kind of groupware uses an enormous

database containing work records, memos, and notations and combines it with a messaging (E-mail) system. It is information-centered and allows people to do workgroup computing, focusing on the information being processed. Thus, a company like accounting giant Coopers & Lybrand uses Lotus Notes software to let co-workers organize and share financial and tax information. It can also be used to relay advice from outside specialists, speeding up audits and answers to complex questions from clients. Groupware is more than just multiuser software, which allows users on a network to access the same data; groupware does this but also allows users to coordinate and keep track of an ongoing project.

- Workflow software: Workflow software, exemplified by ActionWorkflow System and ProcessIt, helps workers understand and redesign the steps that make up a particular process — thus, it is process-centered. It governs the tasks performed and coordinates the transfer of the information required to carry out the tasks. It also routes work automatically among employees and helps organizations reduce paper-jammed bureaucracies.
- Meeting software: Examples of meeting software are Microsoft NetMeeting, Netscape's Collaborator, and Ventana's GroupSystems Ⅴ, which allow people to have computer-linked meetings. With this software, people "talk", or communicate, with one another at the same time by typing on microcomputer keyboards.
- Scheduling software: Scheduling software such as Microsoft Outlook, Microsoft schedulePlus, and Powercore's Network Scheduler 3 uses a microcomputer network to coordinate co-workers' electronic datebooks or appointment calendars so they can figure out a time when they can all get together. (Note: Scheduling software is useful only if everyone uses it regularly and consistently — otherwise appointment information and the like will be missing.)

Groupware has changed the kind of behavior required for success in an organization. For one thing, it requires workers to take more responsibility. Ethically, of course, when you are contributing to a group project of any kind, you should try to do your best. However, when your contribution to the project is clearly visible to all, as happens with groupware, you have to do your best. In addition, using E-mail or groupware means you need to use good manners and be sensitive to others while you're online.

7.3.5 Desktop Accessories

Pretend you are sitting at desk in an old-fashioned office. You have a calendar, clock, calculator, Rolodex-type address file, and notepad. Most of these items could also be found on a student's desk. How would a computer and software improve on this arrangement? Many people find ready uses for types of software known as desktop accessories and personal information managers (PIMs).

- Desktop accessories: A desktop accessory, or desktop organizer, is a software package that provides an electronic version of tools or objects commonly found on a desktop: calendar, clock, card file, calculator, and notepad. Some desktop accessory programs come as standard equipment with systems software (such as Microsoft Windows). Others, such as

Borland's SideKick or Lotus Agenda, are available as separate programs to run in your computer's main memory while you are running other software. Some are principally scheduling and calendaring programs; their main purpose is to enable you to do time and event scheduling. Suppose, for example, you are working on a word processing document and someone calls to schedule lunch next week. You can simply enter a command that "pops up" your appointment calendar, type in the appointment, save the information, and then return to your interrupted work. Other features, such as a calculator keypad, a scratch pad for typing in notes to yourself, and a Rolodex-type address and phone directory (some with an automatic telephone dialer), can be displayed on the screen when needed.

- Personal information managers: A more sophisticated program is the personal information manager (PIM), a combination word processor, database, and desktop accessory program that organizes a variety of information. Examples of PIMs are Commerce, Ecco, and Lotus Organizer. (PIMs are often integrated into E-mail and groupware products).

7.3.6 Web Browsers

The Internet, that network of millions of interconnected networks, "is just a morass of data, dribbling out of [computers] around the world," says one writer. "It is unfathomably chaotic, mixing items of great value with cyber-trash." This is why browser software has caught people's imaginations, he states. "A browser cuts a path through the tangled growth and even creates a form of memory, so each path can be retraced."

The most exciting part of the Internet is probably that fast-growing region or subset of it known as the World Wide Web. The World Wide Web, or simply the Web, consists of hundreds of thousands of intricately interlinked sites — called home pages — set up for on-screen viewing in the form of colorful magazine-style "pages" with text, image, and sound.

To be connected to the World Wide Web, you need a modem and an automatic setup with an online service or Internet access provider, which often provides the browser software for exploring the Web. A Web browser, or simply browser, is software that enables you to "browse through" and view Web sites. You can move from page to page by clicking on or selecting a hyperlink — either underlined text or a graphic, or by typing in the address of the destination page.

There are a great many browsers, including some unsophisticated ones offered by Internet access providers and some by the large commercial online services such as America online, CompuServe, and Prodigy. However, the recent battle royal for Web browser prominence has been between Netscape, which produces Navigator and Communicator, and Microsoft, which offers Microsoft Explorer.

Technical Notes to the Text

1. groupware，群件。一般认为群件是由 3 部分组成的：小组动态交流功能(team dynamics)、文档管理功能(document management)和应用开发功能(application development)。典型的群件产品是 Lotus 公司的 Lotus Notes。

2. project management，项目管理。项目(project)是指在既定的资源和要求的约束下，为了实现某种目标而开展的任务的集合，是一系列活动有机组合而形成的一个完整过程。项目管理是以项目为对象的系统管理方法，通过一个临时性专门机构的柔性组织，对项目进行高效率的计划、组织、指导和控制，以实现项目全过程的动态管理和项目目标的综合协调与优化。项目管理需要通过一个专门的组织实施。项目管理通过规划资源，从时间、成本、质量、客户关系等方面满足项目目标。

3. macro，宏。宏是一系列组合在一起的 Word 命令和指令，它们形成了一个命令，以实现任务执行的自动化。可以创建并执行宏(宏实际上就是一条自定义的命令)，以替代人工进行的一系列费时而单调的重复性操作，自动完成所需任务。

4. OLE (object linking and embedding)，对象链接和嵌入。可以使用链接对象或嵌入对象将在 Office 程序或支持链接对象和嵌入对象的程序中创建的文件的全部或部分添加到其他文件中。链接和嵌入的主要区别在于数据的存放位置及在将其插入目标文件后的更新方式。

5. workflow software，工作流软件。工作流是一类能够完全或者部分自动执行的过程，它根据一系列过程的规划、文档、信息或任务能够在不同的执行者之间进行传递和执行。

Word Bank to the Text

A. Useful new words

specialty	*n.* 专业
entertainment	*n.* 款待，娱乐
collaboration	*n.* 协作
choreographer	*n.* 舞蹈指导
encyclopedia	*n.* 百科全书
keystroke	*n.* [计]键击，按键
scroll	*v.* 使成卷形
margin	*n.* 边，缘
predetermine	*v.* 预定，预先确定
consolidate	*v.* 巩固
tutorial	*adj.* 导师的，辅导的
prescribe	*v.* 指示，规定
booklet	*n.* 小册子
wysiwyg	*abbr.* 所见即所得
alteration	*n.* 变更，改造
thesaurus	*n.* 分类词汇汇编，词典，字典
imitate	*v.* 模仿，仿效
intersection	*n.* [数]交集，十字路口，交叉点
formula	*n.* 公式，规则
ongoing	*adj.* 正在进行的

enormous	*adj*. 巨大的，庞大的
analytical	*adj*. 分析的，解析的
artwork	*n*. 艺术品，美术品
vice versa	*adv*. 反之亦然
exemplify	*v*. 例证，例示
bureaucracy	*n*. 官僚，官僚作风
accessory	*n*. 附件，零件
morass	*n*. 沼泽，困境
dribbling	*n*. 漏泄
unfathomable	*adj*. 深奥的，难解的
chaotic	*adj*. 混乱的，无秩序的
subset	*n*. [数]子集

B. Useful expressions

have a positive effect on	具有积极的影响
differ in	在……不同
a variety of	多种多样的
a row of	一排，一行
a series of	一系列
have no need for	对……没有需求
perform tasks	执行任务
close to	接近，在……附近
pop up	突然出现，弹出
move around	到处活动
contrast with	和……形成对照
stand for	代表，象征
give sb. options for	让某人选择
vice versa	反之亦然
make up	构成，组成
figure out	计算出，解决
contribute to	有助于，贡献
be sensitive to	对……敏感
work on	致力于
a morass of	凌乱不堪的

C. Technical terms and proper names

applications software	应用软件
word processing	字处理技术
spreadsheet	电子表格

personal finance	个人理财
presentation graphic	演示图形
database manager	数据库管理器
groupware	群件
desktop accessory	桌面辅助工具
browser	浏览器
desktop publishing	桌面印刷
project management	项目管理
computer-aided design and manufacturing (CAD/CAM)	计算机辅助设计/制造
multimedia authoring	多媒体发布
animation	动画
MIDI (musical instrument digital interface)	乐器数字化接口
speech synthesis	语音合成
insertion point	插入点
scroll bar	滚动条
window	窗口
menu bar	菜单条
pull-down menu	下拉式菜单
button	按钮
toolbar	工具条
dialog box	对话框
default value	缺省值，默认值
macro	宏
OLE (object linking and embedding)	对象链接和嵌入
clipboard	剪贴板
column	列
row	行
cell	单元格
cell address	单元格地址
cell pointer	单元格指针
formula	公式
function	函数
bar chart	柱形图
line graph	线图
pie chart	圆饼图
workflow software	工作流软件
personal information manager (PIM)	个人信息管理器
Web browser	网页浏览器
World Wide Web	万维网

homepage 主页

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. ____________ software allows you to create and edit documents.
2. ____________ is an activity of moving quickly upward or downward through the text or other screen display.
3. The software which is the layer of software closest to the user is ____________.
4. Microsoft Excel and Lotus 1-2-3 are the principal ____________ programs.
5. ____________ is software that is used on a network and serves a group of users working together on the same project.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. Word processing and database management system are two most popular applications software. ( )
2. Electronic spreadsheet software enables you to perform "what-if" calculations. ( )
3. A worksheet is composed of fields, records, and files. ( )
4. You can access the World Wide Web using multimedia authoring software. ( )
5. OLE is used to collect information from the user and display helpful messages. ( )

Ⅲ. Match each of the following terms with the appropriate definition.

Cell	Macro	Workflow software
Function	Integrated software	

1. _______ The feature that allows you to use a single keystroke, command, or toolbar button to automatically issue a predetermined series of commands.
2. _______ The place where a row and a column intersect in the spreadsheet software.
3. _______ The built-in formulas that perform common calculations in the spreadsheet software.
4. _______ Software that can help workers understand and redesign the steps that make up a particular process.
5. _______ Software that combine the features of several applications programs — such as word processing, spreadsheet, database, graphics, and communications — into one software package.

Ⅳ. List four types of applications software.

1. ____________________________

2. ______________________________
3. ______________________________
4. ______________________________

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

consolidate	chaotic	tutorial
collaboration	enormous	margin
prescribe	ongoing	imitate

1. Close ________ between the Bank and the Fund is not merely desirable, it is essential.
2. Several small businesses are planning to ________ to form a large powerful company.
3. The doctor ________ a holiday as the best cure for his depression.
4. He beat the other runners by a ________ of ten seconds.
5. The methods of study include lectures, ________ , case studies, and practical sessions.
6. He's very clever at ________ his friends.
7. The oceans began to develop a(n) ________ system of living things, with many diverse forms of life, all dependent on one another.
8. Mullins began to search among the ________ mess of papers on his desk.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "employ" and fill in the sentences with the right word.

Example: Default values are standard settings **employed** by the computer when the user does not specify particular margin widths in your page setup.

1. Applications software employs the ca______ of a computer directly to a task that the user wishes to perform.
2. The international accounting company employs the retired pr______ as an adviser.
3. The little girl employs her free ti______ in sewing.
4. The football team succeeded in employing subtle psychological ta______ and won the game.
5. The police employed fo______ to open the door.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

Applications software is the software designed to help you solve problems specific to business or perform specific business tasks. It is a __1__ of computer software that employs the capabilities of a computer directly and __2__ to a task that the user wishes to perform. This should be __3__ with systems software which is involved in integrating a computer's various

capabilities, but typically does not __4__ apply them in the performance of tasks that benefit the user. The exact definition between systems software __5__ as operating systems and applications software is not precise, however, and is occasionally subject __6__ controversy. An application thus differs from an operating system which runs a computer, a utility which performs maintenance or general-purpose chores, and a programming language __7__ which computer programs are created. __8__ on the work for which it was designed, an application can manipulate text, numbers, graphics, or a combination of these elements. Some application packages offer __9__ computing power by focusing on a single task, such as word processing; others, called __10__ software, offer somewhat less power but include several applications, such as a word processor, a spreadsheet, and a database program.

1. A. subclass	B. class	C. kind	D. type
2. A. thorough	B. thoroughly	C. through	D. although
3. A. compared	B. comparing	C. contrasted	D. contrasting
4. A. indirectly	B. mainly	C. directly	D. necessarily
5. A. such	B. so	C. much	D. in
6. A. to	B. in	C. with	D. under
7. A. for	B. with	C. in	D. on
8. A. Depend	B. To depend	C. Depended	D. Depending
9. A. considerate	B. consider	C. consideration	D. considerable
10. A. whole	B. single	C. integrated	D. mass

Translation

Ⅷ. Translate the following into Chinese.

1. Applications software is the software designed to help you solve problems specific to business or perform specific business tasks.

2. Although applications software packages differ in their use of specific commands and functions, most of them have some features in common.

3. Many applications software programs allow you to copy an item from one document and then paste it into another document or application, or copy an item and place the copy in another part of the same document.

4. The user can do all these manipulations on screen, in “wysiwyg” fashion, before printing out hardcopy.

5. It’s worth noting that word processing programs (and indeed most forms of applications software) come from the manufacturer with default settings.

Chapter 8

An Introduction to Database Systems

Pre-reading Questions

1. What are the major disadvantages in a file-processing system?
2. What are the two levels of data independence? Describe each of them.
3. What are the functions of the structured query language (SQL)?

A database system (DBS)consists of a collection of interrelated data and a set of programs to access those data. A database is a collection of data organized to serve many applications efficiently by centralizing the data and minimizing redundant data. The primary goal of a DBMS (DataBase Management System) is to provide an environment that is both convenient and efficient to use in retrieving and storing database information.

Database systems are designed to manage large bodies of information. The management of data involves both the definition of structures for the storage of information and the provision of mechanisms for the manipulation of information. In addition, the database system must provide for the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.

The importance of information in most organizations, which determines the value of the database, has led to the development of a large body of concepts and techniques for the efficient management of data. In this chapter, we will present a brief introduction to the principles of database systems.

8.1 Purpose of Database Systems

The typical file-processing system is supported by a conventional operating system. Permanent records are stored in various files, and different application programs are written to extract records from, and to add records to, the appropriate files. Before the advent of DBMSs, organizations typically stored information using such systems.

Keeping organizational information in a file-processing system has a number of major disadvantages.

- Data redundancy and inconsistency. Data redundancy is the presence of duplicate data in multiple data files. Since the files and application programs are created by different programmers over a long period, the various files are likely to have different formats and the programs may be written in several programming languages. Moreover, the same information may be duplicated in several places (files). This redundancy leads to higher

storage and access cost. In addition, it may lead to data inconsistency; that is, the various copies of the same data may no longer agree.

- Difficulty in accessing data. The point here is that conventional file-processing environment does not allow needed data to be retrieved in a convenient and efficient manner. More responsive data-retrieval systems must be developed for general use.
- Data isolation. Because data are scattered in various files, and files may be in different format, it is difficult to write new application programs to retrieve the appropriate data.
- Integrity problems. The data values stored in the database must satisfy certain types of consistency constraints. Developers enforce the constraints in the system by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.
- Atomicity problems. A computer system, like any other mechanical or electrical device, is subject to failure. In many applications, it is crucial to ensure that, once a failure has occurred and has been detected, the data are restored to the consistent state that existed prior to the failure. Consider a program to transfer $50 from account A to B. If a system failure occurs during the execution of the program, it is possible that the $50 was removed from account A but was not credited to account B, resulting in an inconsistent database state. Clearly, it is essential to database consistency that either both the credit and debit occur, or that neither occurs. That is, the funds transfer must be atomic — it must happen in its entirety or not at all. It is difficult to ensure this property in a conventional file-processing system.
- Concurrent-access anomalies. So that the overall performance of the system is improved and a faster response time is possible, many systems allow multiple users to update the data simultaneously. In such an environment, interaction of concurrent updates may result in inconsistent data. Consider bank account A, containing $500. If two customers withdraw funds (say $50 and $100 respectively) from account A at about the same time, the result of the concurrent executions may leave the account in an incorrect (or inconsistent) state. Suppose that the programs executing on behalf of each withdrawal read the old balance, reduce that value by the amount being withdrawn, and write the result back. If the two programs run concurrently, they may both read the value $500, and write back $450 and $400, respectively. Depending on which one writes the value last, the account may contain either $450 or $400, rather than the correct value of $350. To guard against this possibility, the system must maintain some form of supervision. Because data may be accessed by many different application programs that have not been coordinated previously, however, supervision is difficult to provide.
- Security problems. Not every user of the database system should be able to access all the data. For example, in a banking system, payroll personnel need to see only that part of the database that has information about the various bank employees. They do not need access

to information about customer accounts. Since application programs are added to the system in an ad hoc manner, it is difficult to enforce such security constraints.

These difficulties, among others, have prompted the development of DBMSs. In what follows, we shall see the concepts and algorithms that have been developed for database systems to solve the problems mentioned. A typical data-processing application stores a large number of records, each of which is fairly simple and small.

8.2 View of Data

A DBMS is a collection of interrelated files and a set of programs that allow users to access and modify these files. A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored and maintained.

8.2.1 Levels of Data Abstraction

For the system to be usable, it must retrieve data efficiently. This concern has led to the design of complex data structures for the representation of data in the database. Since many database systems users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users' interactions with the system:

- Physical level. The lowest level of abstraction describes how the data are actually stored. At the physical level, complex low-level data structures are described in detail.
- Logical level. The next-higher level of abstraction describes what data are stored in database, and what relationships exist among those data. The entire database is thus described in terms of a small number of relatively simple structures. Although implementation of the simple structures at the logical level may involve complex physical-level structures, the user of the logical level does not need to be aware of this complexity. The logical level of abstraction is used by database administrators, who must decide what information is to be kept in the database.
- View level. The highest level of abstraction describes only part of the entire database. Despite the use of simpler structures at the logical level, some complexity remains, because of the large size of the database. Many users of the database system will not be concerned with all this information. Instead, such users need to access only a part of the database. So that their interaction with the system is simplified, the view level of abstraction is defined. The system may provide many views for the same database.

The interrelationship among these three levels of abstraction is illustrated in Figure 8-1.

An analogy to the concept of data types in programming languages may clarify the distinction among levels of abstraction. Most high-level programming languages support the notion of a record type. For example, in a Pascal-like language, we may declare a record as follows:

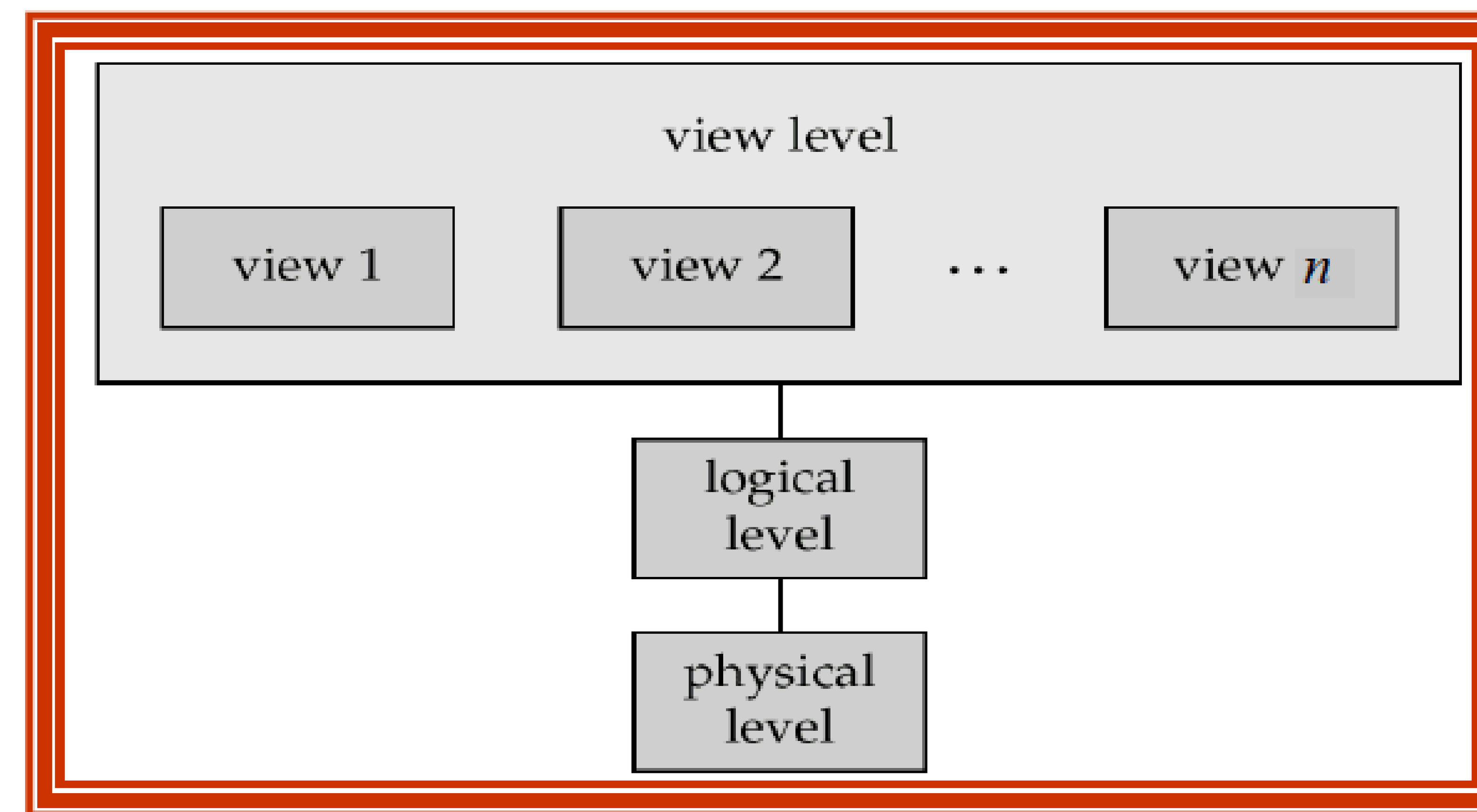


Figure 8-1 The Three Levels of Data Abstraction

```
type customer = record
        customer-name: string;
        social-security: string;
        customer-street: string;
        customer-city: string;
    end;
```

This code defines a new record called customer with four fields. Each field has a name and a type associated with it. A banking enterprise may have several such record types, including:

- account, with fields account-number and balance;
- employee, with fields employee-name and salary.

At the physical level, a customer account, or employee record can be described as a block of consecutive storage locations (for example, words or bytes). The language compiler hides this level of detail from programmers. Similarly, the database system hides many of the lowest-level storage details from database programmers. Database administrators may be aware of certain details of the physical organization of the data.

At the logical level, each such record is described by a type definition, as illustrated in the previous code segment, and the interrelationship among these record types is defined. Programmers using a programming language work at this level of abstraction. Similarly, database administrators usually work at this level of abstraction.

Finally, at the view level, computer users see a set of application programs that hide details of the data types. Similarly, at the view level, several views of the database are defined, and database users see these views. In addition to hiding details of the logical level of the database, the views also provide a security mechanism to prevent users from accessing parts of the database. For example, tellers in a bank see only that part of the database that has information on customer accounts; they cannot access information concerning salaries of employees.

8.2.2 Instances and Schemas

Databases change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an instance of the database. The overall design of

the database is called the database schema. Schemas are changed infrequently, if at all.

Analogies to the concepts of data types, variables, and values in programming languages is useful here. Returning to the customer-record type definition, note that, in declaring the type customer, we have not declared any variables. To declare such variables in a Pascal-like language, we write

```
var customer 1: customer;
```

Variable customer 1 now corresponds to an area of storage containing a customer type record.

A database schema corresponds to the programming-language type definition. A variable of a given type has a particular value at a given instant. Thus, the value of a variable in programming languages corresponds to an instance of a database schema.

Database systems have several schemas, partitioned according to the levels of abstraction that we discussed. At the lowest level is the physical schema; at the intermediate level is the logical schema; and at the highest level is a subschema. In general, database systems support one physical schema, one logical schema, and several subschemas.

8.2.3 Data Independence

The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called data independence. There are two levels of data independence:

- Physical data independence is the ability to modify the physical schema without causing application programs to be rewritten. Modifications at the physical level are occasionally necessary to improve performance.
- Logical data independence is the ability to modify the logical schema without causing application programs to be rewritten. Modifications at the logical level are necessary whenever the logical structure of the database is altered (for example, when money-market accounts are added to a banking system).

Logical data independence is more difficult to achieve than physical data independence, since application programs are heavily dependent on the logical structure of the data that they access.

The concept of data independence is similar in many respects to the concept of abstract data types in modem programming languages. Both hide implementation details from the users, to allow users to concentrate on the general structure, rather than on low-level implementation details.

8.3 Data Models

Underlying the structure of a database is the data model: a collection of conceptual tools for describing data, data relationship, data semantics, and consistency constraints. The various data models that have been proposed fall into three different groups: object-based logical models, record-based logical models, and physical models.

8.3.1 Object-Based Logical Models

Object-based logical models are used in describing data at the logical and view levels. They are characterized by the fact that they provide fairly flexible structuring capabilities and allow data constraints to be specified explicitly. There are many different models, and more are likely to come. Several of the more widely known ones are:

- The entity-relationship model;
- The object-oriented model;
- The semantic data model;
- The functional data model.

The entity-relationship model and the object-oriented model are representatives of the class of the object-based logical models. The entity-relationship model has gained acceptance in database design and is widely used in practice. The object-oriented model includes many of the concepts of the entity-relationship model, but represents executable code as well as data. It is rapidly gaining acceptance in practice. We shall give brief descriptions of both models next.

1. The Entity-Relationship Model

The entity-relationship (E-R) data model is based on a perception of a real world that consists of a collection of basic objects, called entities, and of relationships among these objects. An entity is a "thing" or "object" in the real world that is distinguishable from other objects. For example, each person is an entity, and bank accounts can be considered to be entities. Entities are described in a database by a set of attributes. For example, the attributes account-number and balance describe one particular account in a bank. A relationship is an association among several entities. For example, a depositor relationship associates a customer with each account that she has. The set of all entities of the same type, and the set of all relationships of the same type, are termed an entity set and relationship set respectively.

In addition to entities and relationships, the E-R model represents certain constraints to which the contents of a database must conform. One important constraint is mapping cardinalities, which express the number of entities to which another entity can be associated via a relationship set.

The overall logical structure of a database can be expressed graphically by an E-R diagram, which is built up from the following components:

- Rectangles, which represent entity sets;
- Ellipses, which represent attributes;
- Diamonds, which represent relationships among entity sets;
- Lines, which link attributes to entity sets and entity sets to relationships.

Each component is labeled with the entity or relationship that it represents. As an illustration, consider part of a database banking system consisting of customers and of the accounts that these customers have. The corresponding E-R diagram is shown in Figure 8-2.

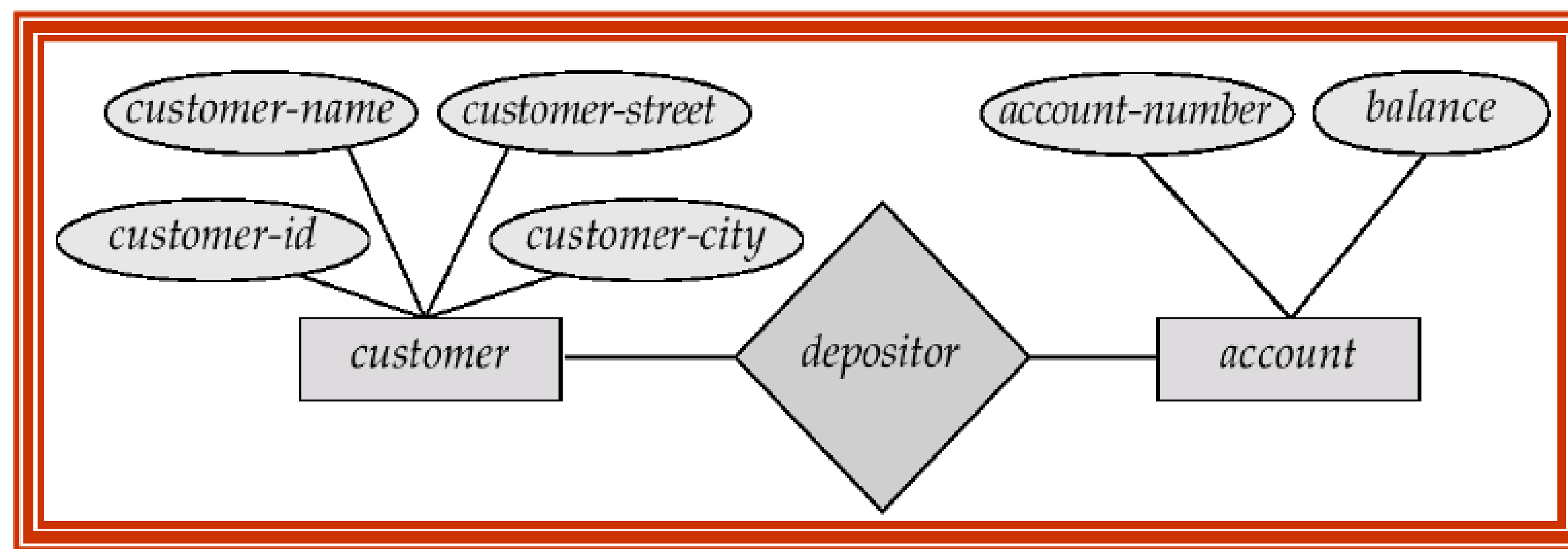


Figure 8-2　A Sample E-R Diagram

2. The Object-Oriented Model

Like the E-R model, the object-oriented model is based on a collection of objects. An object contains values stored instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods.

Objects that contain the same types of values and the same methods are grouped together into classes. A class may be viewed as a type definition for objects. This combination of data and methods comprising a type definition is similar to a programming language abstract data type (ADT).

The only way in which one object can access the data of another object is by invoking a method of that other object. This action is called sending a message to the object. Thus, the call interface of the methods of an object defines that object's externally visible part. The internal part of the object — the instance variables and method code — are not visible externally. The result is two levels of data abstraction.

To illustrate the concept, let us consider an object representing a bank account. Such an object contains instance variables account-number and balance. It contains a pay-interest method, which adds interest to the balance. Assume that the bank had been paying 6 percent interest on all accounts, but now is changing its policy to pay 5 percent if the balance is less than $1000 or 6 percent if the balance is $1000 or greater. Under most data models, making this adjustment would involve changing code in one or more application programs. Under the object-oriented model the only change is made within the pay-interest method. The external interface to the objects remains unchanged.

Unlike entities in the E-R model, each object has its own unique identity, independent of the values that it contains. Thus, two objects containing the values are nevertheless distinct. The distinction among individual objects is maintained in the physical level through the assignment of distinct object identifiers.

8.3.2　Record-Based Logical Models

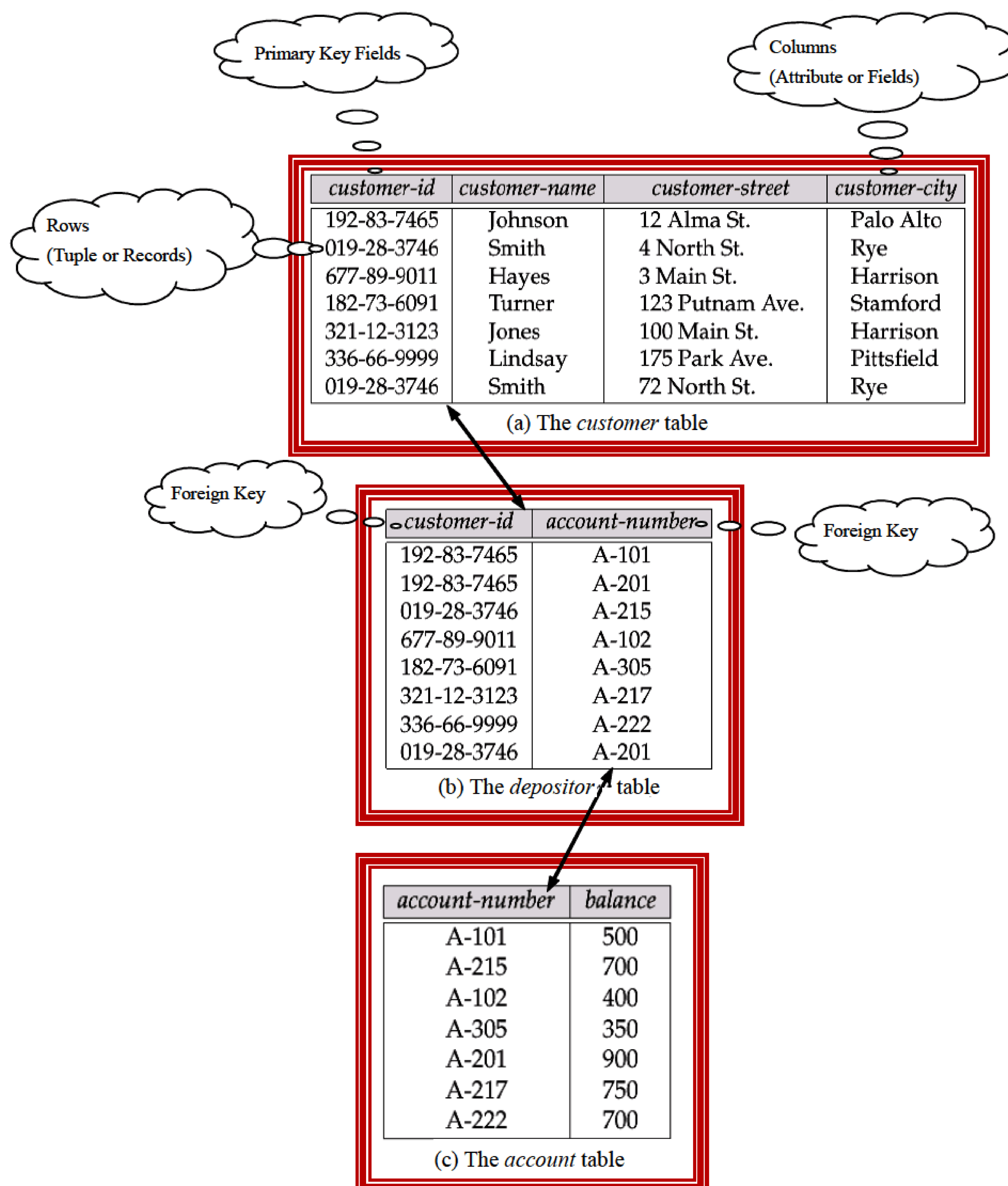
Record-based logical models are used in describing data at the logical and view levels. In contrast to object-based data models, they are used both to specify the overall logical structure of the database and to provide a higher-level description of the implementation.

Record-based models are so named because the database is structured in fixed-format records of several types. Each record type defines a fixed number of fields, or attributes, and each field is

usually of a fixed length. The use of fixed-length records simplifies the physical-level implementation of the database. This simplicity is in contrast to many of the object-based models, whose richer structure often leads to variable-length records at the physical level.

The three most widely accepted record-based data models are the relational, network, and hierarchical models. Next we examine the relational model that has widespread use on personal computers, minicomputers and mainframe systems.

The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. The relational data model represents all data in the database as simple two-dimensional tables called relations. It can relate data stored in one table to data in another as long as the two tables share a common data element. Figure 8-3 presents a sample relational database comprising of two tables: one shows bank customers, and the other shows the accounts that belong to those customers.



customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(b) The *depositor* table

account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(c) The *account* table

Figure 8-3 Example of Tabular Data in the Relational Model

In each table the rows are unique records and the columns are fields. Another term for a row or record in a relation is a tuple.

A super key is a set of one or more attributes that, taken collectively, allows us to identify uniquely an entity (record) in a relation. For example, the customer-id attribute of the relation customer is sufficient to distinguish one customer record from another. Thus, customer-id is a super key. Similarly, the combination of customer-id and customer-name is a super key for the relation customer. The customer-name attribute of customer is not a super key, because several people might have the same name.

The concept of a super key is not sufficient for our purpose, since, as we saw, a super key may contain extraneous attributes. If K is a super key, then so is any superset of K. We are often interested in super keys for which no proper subset is a super key. Such minimal super key is called candidate key. For example, the combination of customer-id and customer-name is a super key but not a candidate key for the relation customer. Customer-id is a candidate key for the relation customer. We shall use the term primary key to denote a candidate key that is chosen by the database designer as the principal means of identifying records within a relation.

The strengths of relational DBMS are great flexibility in regard to ad hoc queries, power to combine information from different sources, simplicity of design and maintenance, and the ability to add new data and records without disturbing existing programs and applications. However, these systems are somewhat slower because they typically require many accesses to the data stored on disk to carry out the select, join, and project commands.

8.3.3 Physical Data Models

Physical data models are used to describe data at the lowest level. In contrast to logical data models, there are few physical data models in use. Two of the widely known ones are the unifying model and the frame-memory model.

8.4 Database Languages

A database system provides two different languages: one to specify the database schema, and the other to express database queries and updates.

8.4.1 Data Definition Language (DDL)

A database schema is specified by a set of definitions expressed by a special language called a data definition language. The result of compilation of DDL statements is a set of tables that is stored in a special file called data dictionary, or data directory.

A data dictionary is a file that contains metadata — that is, data about data. This file is consulted before actual data are read or modified in the database system.

The storage structure and access methods used by the database system are specified by a set

of definitions in a special type of DDL called a data storage and definition language. The result of compilation of these definitions is a set of instructions to specify the implementation details of the database schemas — details are usually hidden from the users.

8.4.2 Data Manipulation Language (DML)

The levels of abstraction that we discussed in Section 8.2 apply not only to the definition or structuring of data, but also to the manipulation of data. By data manipulation, we mean:

- The retrieval of information stored in the database;
- The insertion of new information into the database;
- The deletion of information from the database;
- The modification of information stored in the database.

At the physical level, we must define algorithms that allow efficient access to data. At higher levels of abstraction, we emphasize ease of use. The goal is to provide efficient human interaction with the system.

A data manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model. There are basically two types:

- Procedural DMLs require a user to specify what data are needed and how to get those data.
- Nonprocedural DMLs require a user to specify what data are needed without specifying how to get those data.

Nonprocedural DMLs are usually easier to learn and use than procedural DMLs. However, since a user does not have to specify how to get the data, these languages may generate code that is not as efficient as that produced by procedural languages. We can remedy this difficulty through various optimization techniques.

A query is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a query language. Although technically incorrect, it is common practice to use the terms query language and data-manipulation language synonymously.

8.4.3 The Structured Query Language (SQL)

The structured query language is the most widely used and standard query language for relational database management systems. It is a kind of nonprocedural language. The SQL language has several parts:

- Data-definition language (DDL). The SQL DDL provides commands for defining relation schemas, deleting relations, creating indices, and modifying relation schemas.
- Interactive data-manipulation language (DML). The SQL DML includes a query language based on both the relational algebra and the tuple relational calculus. It includes also commands to insert tuples into, delete tuples from, and modify tuples in the database.
- Embedded DML. The embedded form of SQL is designed for use within general-purpose programming languages, such as PL/1, COBOL, Pascal, FORTRAN and C.

- View definition. The SQL DDL includes commands for defining views.
- Authorization. The SQL DDL includes commands for specifying access rights to relations and views.
- Integrity. The SQL DDL includes commands for specifying integrity constraints that the data stored in the database must satisfy. Updates that violate integrity constraints are disallowed.
- Transaction control. SQL includes commands for specifying the beginning and ending of transactions. Several implementations also allow explicit locking of data for concurrency control.

8.5 Transaction Management

Often, several operations on the database form a single logical unit of work. An example that we saw earlier is a fund transfer, in which one account (say A) is debited and another account (say B) is credited. Clearly, it is essential that either both the credit and debit occur, or that neither occurs. That is, the fund transfer must happen in its entirety or not at all. This all-or-none requirement is called atomicity. In addition, it is essential that the execution of the fund transfer preserve the consistency of the database. That is, the value of the sum A + B must be preserved. This correctness requirement is called consistency. Finally, after the successful execution of a fund transfer, the new values of accounts A and B must persist, despite the possibility of system failures. This persistency requirement is called durability.

A transaction is a collection of operations that performs a single logical function in a database application. Each transaction is a unit of both atomicity and consistency. Thus, we require that transaction do not violate any database consistency constraints. That is, if the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates. However, during the execution of a transaction, it may be necessary temporarily to allow inconsistency. This temporary inconsistency, although necessary, may lead to difficulty if a failure occurs.

It is the responsibility of the programmer to define properly the various transactions, so that each preserves the consistency of the database. For example, the transaction to transfer funds from account A to account B could be defined to be composed of two separate programs: one that debits account A, and the other that credits account B. The execution of these two programs one after the other will indeed preserve consistency. However, each program by itself does not transform the database from a consistent state to a new consistent state. Thus, those programs are not transactions.

Ensuring the atomicity and durability properties is the responsibility of the database system itself — specifically, of the transaction-management component. In the absence of failures, all transactions complete successfully, and atomicity is achieved easily. However, due to various

types of failures, a transaction may not always complete its execution successfully. If we are to ensure the atomicity property, a failed transaction must have no effect on the state of the database. Thus, the database must be restored to the state in which it was before the transaction in question started executing. It is the responsibility of the database system to detect system failures and to restore the database to a state that existed prior to the occurrence of the failure.

Finally, when several transactions update the database concurrently, the consistency of data may no longer be preserved, even though each individual transaction is correct. It is the responsibility of the concurrency control manager to control the interaction among the concurrent transactions, to ensure the consistency of the database.

Database systems designed for use on small personal computers may not have all the features noted. For example, many small systems impose the restriction of only one user being allowed to access the database at a time. Others leave the tasks of backup and recovery to the user. This setup allows for a smaller data manager, with fewer requirements for physical resources — especially main memory. Although such a low-cost, low-feature approach is sufficient for small personal databases, it is inadequate to meet the needs of a medium-to-large-scale enterprise.

8.6 Database Administrator

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. The person who has such central control over the system is called the database administrator (DBA). The functions of the DBA include the following:

- Schema definition. The DBA creates the original database schema by writing a set of definitions that is translated by the DDL compiler to a set of tables that is stored permanently in the data dictionary.
- Storage structure and access-method definition. The DBA creates appropriate storage structures and access methods by writing a set of definitions, which is translated by the data-storage and data-definition-language compiler.
- Schema and physical-organization modification. Programmers accomplish the relatively rare modifications either to the database schema or to the description of the physical storage organization by writing a set of definitions that is used by either the DDL compiler or the data-storage and data-definition-language compiler to generate modifications to the appropriate internal system tables (for example, the data dictionary).
- Granting of authorization for data access. The granting of different types of authorization allows the database administrator to regulate which parts of the database various users can access. The authorization information is kept in a special system structure that is consulted by the database system whenever access to the data is attempted in the system.
- Integrity-constraint specification. The data values stored in the database must satisfy certain consistency constraints. For example, perhaps the number of hours an employee

may work in one week may not exceed a specified limit (say, 80 hours). Such a constraint must be specified explicitly by the database administrator. The integrity constraints are kept in a special system structure that is consulted by the database system whenever an update takes place in the system.

Technical Notes to the Text

1. database，数据库。数据库是长期存储在计算机内的、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述、存储，具有较小的冗余度、较高的数据独立性和易扩展性，并可为各种用户共享。

2. database system，数据库系统。其包括数据库、数据库管理系统、数据库管理员(DBA)，以及有关的硬件与软件系统。

3. database management systems (DBMS)，数据库管理系统。数据库管理系统是为数据库的建立、使用和维护而配置的软件，它建立在操作系统的基础上，对数据库进行统一的管理和控制。

4. data dictionary (DD)，数据字典。数据库系统中存放的 3 级结构定义的数据库称为数据字典。数据字典也称为系统目录(system catalog)或元数据(metadata)。

5. schema，模式。一组以数据定义语言来表达的语句集，该语句集完整地描述了数据库的结构。

6. physical schema，物理模式。它是最靠近存储设备的存储模式，反映了数据库的存储方式和物理结构。

7. logical schema，逻辑模式，也称概念模式(conceptual schema)。其表示整个数据库的全部信息的结构特征，也被称为全局视图。全局视图表示了数据库的整体逻辑关系。

8. subschema，子模式，也称外模式(external schema)。其是最接近于用户的一层，是用户的数据视图或外视图，程序员看到和使用的数据库的每一个子集称为用户的应用视图。

9. data independence，数据独立性。其指在某个层次上修改模式的定义而不影响位于其上一层模式的能力。

10. physical data independence，物理数据独立性。其指修改物理模式而不必重写应用程序的能力。

11. logical data independence，逻辑数据独立性。其指修改逻辑模式而不必重写应用程序的能力。

12. data model，数据模型。数据模型是描述数据、数据之间的联系、数据语义，以及一致性约束的概念工具的集合。

13. entity-relationship model，实体-联系模型，也称 E-R 模型。E-R 模型是 P.P.Chen 于 1976 年提出的。该模型是直接从现实世界中抽象出实体类型及实体间的联系，然后用 E-R 图表示的数据模型。

14. entity，实体。实体是现实世界中可区别于其他对象的“事件”或“物体”。

15. entity set，实体集。具有相同类型及相同性质(或属性)的实体的集合。

16. mapping cardinality，映射基数。在 E-R 模型中，描述一个实体通过联系集最多或最少能与其他多少个实体相联系的约束称为映射基数。

17. attribute，属性。实体所具有的性质称为属性。

18. relation，关系。关系即满足一定条件的二维表。

19. tuple，元组。元组指关系中的行或记录。

20. super key，超键(码)。超键能唯一标识每个实体的一个属性或几个属性的组合。

21. candidate key，候选键(码)。候选键是最小的超键，其任意真子集都不能成为超键。

22. primary key，主键(码)。主键也称主关键字，是使用频率最高的一个候选键。

23. foreign key，外键(码)。当关系中某个属性(或属性组)虽然不是该关系的主键或只是主键的一部分，但却是另一个关系的主键时，称该属性(或属性组)为这个关系的外键。外键的作用是联系两个关系。

24. data definition language (DDL)，数据定义语言。定义数据库的 3 级结构，包括外模式、概念模式、内模式及其相互之间的映像，定义数据的完整性、安全控制等约束。

25. data manipulation language (DML)，数据操纵语言。实现对数据库中数据的操作。基本的数据操作分成两类：检索(查询)和更新(插入、删除、修改)。

26. procedure DML，过程化 DML。用户编程时，不仅需要指出“做什么”(需要什么样的数据)，还需要指出“怎么做”(怎样获得这些数据)。

27. non-procedure DML，非过程化 DML。用户编程时，只需要指出“做什么”，不需要指出“怎么做”。

28. structured query language (SQL)，结构化查询语言。SQL 在 1970 年由美国 IBM 研究中心的 E.F.Codd 发表的论文中提出，1974 年 Boyce 和 Chamberlin 把 SQUARE 语言改为 SEQUEL 语言，现在其还在不断完善和发展之中。SQL 虽然名为查询，但实际上具有定义、查询、更新和控制等多种功能。

29. view，视图。视图是从一个或几个基本表(或其他视图)导出的表，它与基本表不同，是一个虚表，数据库中只存放视图的定义，而不存放视图对应的数据。

30. atomicity，原子性。事务所有的操作在数据库中要么全部正确反映出来，要么全部不反映，即事务中所包含的操作要么全部执行，要么全部不执行。

31. consistency，一致性。一个事务独立执行的结果将保证数据库的一致性，即数据不会因为事务的执行而遭受破坏。

32. duration，持久性。一个事务成功完成后，它对数据库的改变必须是永久的，即使系统出现故障也不例外。

33. transaction，事务。事务是由一个或多个 SQL 语句系列聚集起来的一个数据库环境中不可分割的逻辑工作单元。事务中每一条语句都是执行某个任务的一部分，但是它们联合起来能够完成某个特定的任务。事务中所有语句必须要么全部执行，要么全部不执行，从而保证数据库的一致状态。

34. relational algebra，关系代数。关系模型建立在集合代数的基础上。关系模型中常用的关系代数操作包括并、交、差、笛卡尔乘积、选择、投影、连接、除。

35. database administrator (DBA)，数据库管理员。数据库管理员负责定义数据库中信息的内容和结构、确定数据库的存储结构和存取策略、定义数据的安全性和完整性约束条件、

监控数据库的使用和运行、定期对数据库进行重组和重构等工作。

Word Bank to the Text

A. Useful new words

interrelate	*v*. (使)相互关联
provision	*n*. 供应，预备
manipulation	*n*. 处理，操作
crash	*n*. 碰撞，崩溃
unauthorized	*adj*. 未被授权的，未经认可的
anomalous	*adj*. 不规则的，反常的
advent	*n*. 到来，来临
redundancy	*n*. 冗余
inconsistency	*n*. 矛盾
duplicate	*adj*. 复制的，两重的
isolation	*n*. 隔绝，孤立，隔离
integrity	*n*. 完整
constraint	*n*. 约束，强制
atomicity	*n*. 原子数
crucial	*adj*. 极重要的
anomaly	*n*. 不规则，异常，反常
respectively	*adv*. 分别地，各个地
withdrawal	*n*. 收回，撤回
supervision	*n*. 监督，管理
payroll	*n*. 薪水册
distinguishable	*adj*. 可区别的，可辨识的
abstraction	*n*. 提取
distinction	*n*. 区别，差别
teller	*n*. (银行)出纳员
schema	*n*. 计划，方案
intermediate	*adj*. 中间的　*n*. 媒介
invoke	*v*. 调用
balance	*n*. 结余，余额
relational	*adj*. 有联系的，有关系的
extraneous	*adj*. 无关系的，外来的
remedy	*v*. 补救，矫正
disallow	*v*. 不准许，禁止

transaction	*n.* 交易，事务

B. Useful expressions

lead to	导致
a large body of	大量的
be subject to	易受……的，受控制
prior to	先于，在……以前
transfer…from…to…	从……迁移到……
be credited to	相信，信赖
withdraw from …	从……提取，取回
on behalf of	作为……的代理，在……一边
guard against	警惕
concentrate on	集中，全神贯注于
be characterized by	具有……特征
in practice	在实践中，实际上
be distinguishable from	可与……相区别
distinguish from	区分，辨别
be sufficient for	充分的，足够的
conform to	符合，遵照
in regard to	关于
apply to	将……应用于
have a control over	对……控制
exceed the limit of	超出限制

C. Technical terms and proper names

database management systems (DBMS)	数据库管理系统
instance	实例
schema	模式
physical schema	物理模式，存储模式，内模式
logical schema	逻辑模式，概念模式，模式
subschema	子模式，外模式
data independence	数据独立性
physical data independence	物理数据独立性
logical data independence	逻辑数据独立性
data model	数据模型
entity-relationship model	实体-联系模型
object-oriented model	面向对象模型
semantic data model	语义数据模型
functional data model	功能数据模型

entity	实体
entity set	实体集
mapping cardinality	映射基数
abstract data type	抽象数据类型
attribute	属性
relation	关系
tuple	元组
primary key	主键
super key	超键
candidate key	候选键
foreign key	外键
data definition language (DDL)	数据定义语言
data dictionary	数据字典
data manipulation language (DML)	数据操纵语言
procedure DML	过程化 DML
non-procedure DML	非过程化 DML
structured query language (SQL)	结构化查询语言
view	视图
the relational algebra	关系代数
the tuple relational calculus	元组关系演算
atomicity	原子性
consistency	一致性
duration	持久性
transaction	事务
database administrator (DBA)	数据库管理员

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. A special file in the DBMS called the ________ maintains descriptions of the structure of data used in the database.

2. A(n) ________ is a collection of multiple related files that are created and managed by a DBMS.

3. A(n) ________ coordinates all activities related to an organization's database.

4. The collection of information stored in the database at a particular moment is called an ________ of the ________.

5. The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called data independence. There are two levels of data independence ________ and ________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The primary goal of a DBMS is to provide an environment that is both convenient and efficient to use in retrieving and storing database information. ( )

2. Ensuring backup and recovery of a database is not one of the functions of a database administrator. ( )

3. The use of key fields makes it easier to locate a record in a database. ( )

4. Physical data independence is more difficult to achieve than logical data independence. ( )

5. Data redundancy may lead to higher storage and access cost. In addition, it may lead to data inconsistency. ( )

Ⅲ. Match each of the following terms with the appropriate definition.

data dictionary	data independence	data model
database management system	data-manipulation language (DML)	

1. ________ A collection of interrelated files and a set of programs that allow users to access and modify these files.

2. ________ A collection of conceptual tools for describing data, data relationship, data semantics, and consistency constraints.

3. ________ The ability to modify a schema definition in one level without affecting a schema definition in the next higher level.

4. ________ A file that contains metadata — that is, data about data.

5. ________ A language that enables users to access or manipulate data as organized by the appropriate data model.

Ⅳ. List three levels of data abstraction.

1. ______________________________
2. ______________________________
3. ______________________________

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

withdrawal	constraint	respectively
remedy	duplicate	manipulation
distinction	integrity	redundancy

1. His clever ________ of the stock markets makes him lots of money.

2. Thousands of bank employees are facing ________ as their employers cut costs.

3. Scientists hope the work done in collaboration with other researchers may be ________ elsewhere.

4. Separatist movements are a threat to the ________ of the nation.

5. There are many different kinds of natural ________ to help overcome winter infections.

6. I did not yet make a ________ between the pleasures of reading and of writing fiction.

7. If you experience any unusual symptoms after ________ of the treatment then contact your doctor.

8. There are no ________ on your choice of subject for the essay.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "invoke" and fill in the sentences with the right word.

Example: The only way in which one object can access the data of another object is by **invoking** a **method** of that other object.

1. The judge invoked an international la______ that protects refugees.

2. In political matters George Washington went out of his way to avoid invoking the au______ of Christ.

3. "Appalachian Spring" by Aaron Copland invoked the at______ of the wide open space of the grassland.

4. The great magicians of old always invoked their go______ with sacrifice.

5. He invoked the na______ of Freud in support of his argument.

6. The ar______ he had invoked when he rejected our bail motion.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

Database systems are designed to manage large bodies of information. The management of data involves both the definition of structures for the storage of information __1__ the provision of mechanisms for the manipulation of information. In __2__, the database system must provide for the safety of the information stored, despite system crashes or attempts at __3__ access. The importance of information in most organizations, __4__ determines the value of the database, has led to the development of a large body of concepts and techniques for the efficient management of data.

The four most __5__ types of organizations are the hierarchical, network, relational and object models. The __6__ structure depends on the natural organization of the application's data, and __7__ the application's requirements which include transaction rate (speed), reliability, maintainability, scalability, and cost.

The strengths of relational DBMS are great flexibility in __8__ to ad hoc queries, power to combine information from different sources, simplicity of design and maintenance, and the __9__ to add new data and records without disturbing existing programs and applications. However, these systems are __10__ slower because they typically require many accesses to the data stored on disk to carry out the select, join, and project commands.

1. A. or	B. as well as	C. and	D. yet
2. A. add	B. addition	C. adding	D. additive
3. A. illegal	B. unauthorized	C. legal	D. authorized
4. A. which	B. what	C. how	D. why
5. A. common	B. good	C. better	D. best
6. A. good	B. optimal	C. option	D. optional
7. A. in	B. about	C. on	D. with
8. A. about	B. with	C. regard	D. over
9. A. able	B. capacity	C. capability	D. ability
10. A. how	B. what	C. somehow	D. somewhat

Translation

Ⅷ. Translate the following into Chinese.

1. Since the files and application programs are created by different programmers over a long period, the various files are likely to have different formats and the programs may be written in several programming languages.

2. In what follows, we shall see the concepts and algorithms that have been developed for database systems to solve the problems mentioned.

3. Despite the use of simpler structures at the logical level, some complexity remains, because of the large size of the database.

4. Returning to the customer-record type definition, note that, in declaring the type customer, we have not declared any variables.

5. Ensuring the atomicity and durability properties is the responsibility of the database system itself — specifically, of the transaction-management component.

Chapter 9
Software Engineering

Pre-reading Questions

1. How many principal functions does software have? Describe each of them.

2. What is software engineering?

3. How many phases are usually involved in software development? Describe the tasks of each phase of them.

9.1 What Is Software Engineering?

What is software? Software refers to the detailed instructions that control the operation of computer hardware. Without the instructions provided by software, computer hardware is unable to perform any of the tasks we associate with computers. Software has three principal functions:

- Manage the computer resources of the organization;
- Provide tools for human beings to take advantage of these resources;
- Act as an intermediary between organizations and stored information.

The phrase software engineering was coined in 1968 as a statement of aspiration — a sort of rallying cry. That year NATO convened a workshop by that name to assess the state and prospects of software production. Capturing the imagination of software developers, the phrase achieved popularity during the 1970s. It now refers to a collection of management processes, software tooling, and design activities for software development. The resulting practice, however, differs significantly from the practice of older forms of engineering.

According to the IEEE *Standard Computer Dictionary* (1990), software engineering is the application of a systematic, disciplined, quantifiable approach to development, operation, and maintenance of software; that is, the application of engineering to software. The aim of software engineering is the production of quality software, delivered on time, within budget, and satisfying users' needs.

As software engineers, we use our knowledge of computers and computing to help solve problems. To help us solve a problem, we employ a variety of methods, tools, procedures, and paradigms.

9.2 Key Issues of Software Engineering

There are eight fundamental notions in software engineering that form the basis for an

effective discipline of software engineering. We introduce them briefly here.

1. Abstraction

Sometimes, looking at a problem in its "natural state" (i.e., as expressed by the customer or user) is a daunting task. We cannot see an obvious way to tackle the problem in an effective or even feasible way. An abstraction is a description of the problem at some level of generalization that allows us to concentrate on the key aspects of the problem without getting mired in the details. This notion is different from a transformation, where we translate the problem to another environment that we understand better; transformation is often used to move a problem from the real world to the mathematical world, so we can manipulate numbers to solve the problem.

Typically, abstraction involves identifying classes of objects that allow us to group items together; this way, we can deal with fewer things and concentrate on the commonalities of the items in each class. We can talk of the properties or attributes of the items in a class and examine the relationships among properties and classes.

2. Analysis and Design Methods and Notations

When you design a program as a class assignment, you usually work on your own. The documentation that you produce is a formal description of your notes to yourself about why you chose a particular approach, what the variable names mean, and which algorithm you implemented. But when you work with a team, you must communicate with many other participants in the development process. Most engineers, no matter what kind of engineering they do, use a standard notation to help them communicate, and to document decisions. For example, an architect draws a diagram or blueprint that any other architect can understand. More importantly, the common notation allows the building contractor to understand the architect's intent and ideas.

Analysis and design methods offer us more than a communication medium. They allow us to build models and check them for completeness and consistency. Moreover, we can more readily reuse requirements and design components from previous projects, increasing our productivity and quality with relative ease.

But there are many open questions to be resolved before we can settle on a common set of methods and tools. Different tools and techniques address different aspects of a problem, and we need to identify the modeling primitives that will allow us to capture all important aspects of a problem with a single technique. Or we need to develop a representation technique that can be used with all methods, possibly tailored in some way.

3. User Interface Prototyping

Prototyping means building a small version of a system, usually with limited functionality that can be used to:

- Help the user or customer identify the key requirements of a system.
- Demonstrate feasibility of a design or approach.

Often, the prototyping process is iterative: We build a prototype, evaluate it (with user and customer feedback), consider how changes might improve the product or design, and then build another prototype. The iteration ends when we and our customers think we have a satisfactory solution to the problem at hand.

Prototyping is often used to design a good user interface: the part of the system with which the user interacts. However, there are other opportunities for using prototypes, even in embedded systems (i.e., in systems where the software functions are not explicitly visible to the user). The prototype can show the user what functions will be available, regardless of whether they are implemented in software or hardware. Since the user interface is, in a sense, a bridge between the application domain and the software development team, prototyping can bring to the surface issues and assumptions that may not have been clear using other approaches to requirements analysis.

4. Software Architecture

The overall architecture of a system is important not only to the ease of implementing and testing it, but also to the speed and effectiveness of maintaining and changing it. The quality of the architectures can make or break a system.

A system's architecture describes the system in terms of a set of architectural units, and a map of how the units relate to one another. The more independent the units are, the more modular the architecture is and the more easily we can design and develop the pieces separately. Wasserman (1996) points out that there are at least five ways that we can partition the system into units:

- Modular decomposition: based on assigning functions to modules.
- Data-oriented decomposition: based on external data structures.
- Event-oriented decomposition: based on events that the system must handle.
- Outside-in design: based on user inputs to the system.
- Object-oriented design: based on identifying classes of objects and their interrelationships.

These approaches are not mutually exclusive. For example, we can design a user interface with event-oriented decomposition while we design the database using object-oriented or data-oriented design. The importance of these approaches is their capture of our design experience, enabling us to capitalize on our past projects by reusing both what we have done and what we have learned by doing it.

5. Software Process

Since the late 1980s, many software engineers have paid careful attention to the process of developing software, as well as to the products that result. The organization and discipline in the activities have been acknowledged to contribute to the quality of the software and to the speed with which it is developed. However, Wasserman notes that: the great variations among application types and organizational cultures make it impossible to be prescriptive about the process itself. Thus, it appears that the software process is not fundamental to software engineering in the same way as are abstraction and modularization.

Instead, he suggests that different types of software need different processes. In particular, Wasserman suggests that enterprisewide applications need a great deal of control, whereas individual and departmental applications can take advantage of rapid application development.

By using today's tools, many small and medium-sized systems can be built by one or two developers, each of whom must take on multiple roles. The tools may include a text editor, programming environment, testing support, and perhaps a small database to capture key data elements about the products and processes themselves. Because the project's risk is relatively low, little management support or review is needed.

However, large, complex systems need more structure, checks, and balances. These systems often involve many customers and users, and development continues over a long period of time. Moreover, the developers do not always have control over the entire development, as some critical subsystems may be supplied by others or be implemented in hardware. This type of high-risk system requires analysis and design tools, project management, configuration management, more sophisticated testing tools, and a more rigorous system of review and causal analysis.

6. Reuse

In software development and maintenance, we often take advantage of the commonalities across applications by reusing items from previous development. For example, we use the same operating system or database management system from one development project to the next, rather than building a new one each time. Similarly, we reuse sets of requirements, parts of designs, and groups of test scripts or data when we build systems that are similar to but not the same as what we have done before. Barnes and Bollinger (1991) point out that reuse is not a new idea, and they provide many interesting examples of how we reuse much more than just code.

Prieto-Diaz (1991) introduced the notion of reusable components as a business asset. Companies and organizations invest in items that are reusable and then gain quantifiable benefit when those items are used again in subsequent projects. However, establishing a long-term, effective reuse program can be difficult, because there are several barriers:

- It is sometimes faster to build a small component than to search for one in a repository of reusable components.
- It may take extra time to make a component general enough to be reusable easily by other developers in the future.
- It is difficult to document the degree of quality assurance and testing that have been done, so that a potential reuser can feel comfortable about the quality of the component.
- It is not clear who is responsible if a reused component fails or needs to be updated.
- It can be costly and time-consuming to understand and reuse a component written by someone else.
- There is often a conflict between generality and specificity.

7. Measurement

Improvement is a driving force in software engineering research: improving our processes, resources, and methods so that we produce and maintain better products. But sometimes we express improvement goals generally, with no quantitative description of where we are and where we would like to go. For this reason, software measurement has become a key aspect of good software engineering practice. By quantifying where we can and what we can, we describe our actions and their outcomes in a common mathematical language that allows us to evaluate our progress. In addition, a quantitative approach permits us to compare progress across disparate projects.

8. Tools and Integrated Environments

For many years, vendors touted CASE (Computer-Aided Software Engineering) tools, where standardized, integrated development environments would enhance software development. However, we have seen how different developers use different processes, methods, and resources, so a unifying approach is easier said than done.

On the other hand, researchers have proposed several frameworks that allow us to compare and contrast both existing and proposed environments. These frameworks permit us to examine the services provided by each software engineering environment and to decide which environment is best for a given problem or application development.

One of the major difficulties in comparing tools is that vendors rarely address the entire development life cycle. Instead, they focus on a small set of activities, such as design or testing, and it is up to the user to integrate the selected tools into a complete development environment. Wasserman (1990) has identified five issues that must be addressed in any tool integration:

- Platform integration: the ability of tools to interoperate on a heterogeneous network.
- Presentation integration: commonality of user interface.
- Process integration: linkage between the tools and the development process.
- Data integration: the way tools share data.
- Control integration: the ability for one tool to notify and initiate action in another.

9.3 Software Process

Software development is a both creative and step-by-step process, often involving many people producing many different kinds of products. A process is a series of steps involving activities, constraints, and resources that produce an intended output of some kind. A process usually involves a set of tools and techniques.

Any process has the following characteristics:

- The process prescribes all of the major process activities.
- The process uses resources, subject to a set of constraints (such as schedule), and produces intermediate and final products.

- The process may be composed of subprocesses that are linked in some way. The process may be defined as a hierarchy of processes, organized so that each subprocess has its own process model.
- Each process activity has entry and exit criteria, so that we know when one activity begins and ends.
- The activities are organized in a sequence, so that it is clear when one activity is performed relative to the other activities.
- Every process has a set of guiding principles that explain the goals of each activity.
- Constraints or controls may apply to an activity, resource, or product. For example, the budget or schedule may constraint the length of time an activity may take or a tool may limit the way in which a resource may be used.

When the process involves the building of some product, we sometimes refer to the process as a life cycle. Thus, the software development process is sometimes called the software development life cycle (SDLC), because it describes the life of a software product from its conception to its implementation, delivery, use, and maintenance.

Software development usually involves the following stages:

- Requirements analysis and definition;
- System design;
- Program design;
- Writing the programs (program implementation);
- Unit testing;
- Integration testing;
- System testing;
- System delivery;
- Maintenance.

Each stage is itself a process (or collection processes) that can be described as a set of activities. And each activity involves constraints, outputs, and resources.

1. Requirements Analysis and Definition

A requirement is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose. Figure 9-1 illustrates the process of determining the requirements for a software-based system. First, we work with our customers to elicit the requirements, by asking questions, demonstrating similar systems, or even developing prototype of all or part of the proposed system. Next, we capture those requirements in a document or database, which is called requirements definition document. The requirements are written first so that we and our customers agree on what the system should do. Then, then requirements are often rewritten, usually in a more mathematical representation, so that the designers can transform the requirements into a good system design. A verification step ensures that the requirements are complete, correct, and consistent, and a validation step makes sure that we have described what the

customer intends to see in the final product.

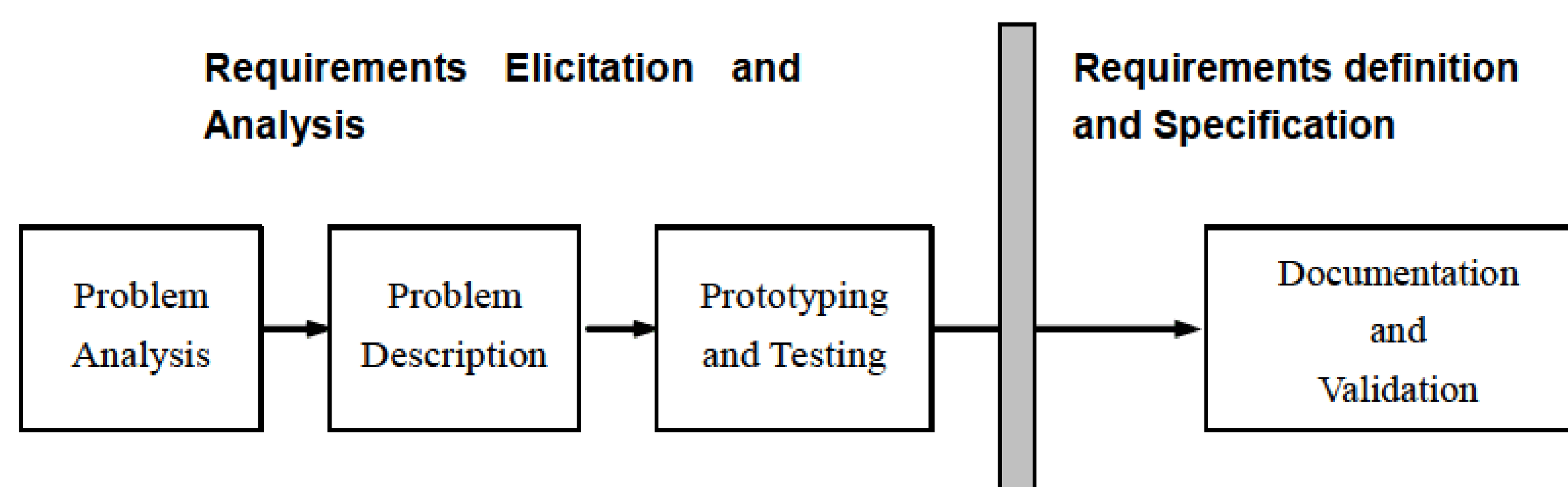


Figure 9-1　The Process of Determining Requirements

2. System Design

Design is the creative process of transforming the problem into a solution; the description of a solution is also called design. To transform requirements into a working system, designers must satisfy both customers and the system builders on our development team. The customers understand what the system is to do. At the same time, the system builders must understand how the system is to work. For this reason, design is really a two-part iterative process. First, we produce a conceptual design or system design that tells the customer exactly what the system will do. Once the customer approves the conceptual design, we translate the conceptual design into a much more detailed document, the technical design, that allows system builders to understand the actual hardware and software needed to solve the customer's problem. The process is iterative because, in actuality, the designers move back and forth among activities involving understanding the requirements, proposing possible solutions, testing aspects of a solution for feasibility, presenting possibilities to the customers, and documenting the design for the programmers.

To design a system is to determine a set of components and component interfaces that satisfy a specified set of requirements. Just as there are many ways to elicit and document the requirements, there are many ways to create good designs. Sometimes the choice is based on designer preferences; other times, the method is dictated by the system's required structure or data. However, every design method involves some kind of decomposition: starting with a high-level depiction of the system's key elements and creating lower-level looks at how the system's features and functions will fit together.

Wasserman suggests that designs are created in one of the five ways:

- Modular decomposition: This construction is based on assigning functions to components. The designers begin with a high-level description of the functions that are to be implemented and build lower-level explanations of how each component will be organized and related to other components.
- Data-oriented decomposition: This design is based on external data structures. The high-level description depicts general data structures, and lower-level descriptions provide detail on what data elements will be involved and how they are related.
- Event-oriented decomposition: This design is based on events that the system must handle

and uses information about how events change the system's state. The high-level description catalogs the various states and lower-level descriptions describe how state transformations take place.

- Outside-in design: This black-box approach is based on user inputs to the system. That is, the high-level description lists all possible inputs a user can make, and then lower-level descriptions address what the system does with each input (including what outputs are produced).
- Object-oriented design: This design identifies classes of objects and their interrelationships. At the highest level, each object type is described. At the lower level, the object attributes and actions are discussed, and the design explains how objects are related to one another.

3. Program Design and Writing the Programs (Program Implementation)

After we understand the users' problem and to devise a high-level solution for it, we must focus on implementing the solution as software. That is, we must write the programs that implement the design.

Programming involves a great deal of creativity. The design is a guide to the function or purpose of each component, but the programmer has great flexibility in implementing the design as code. The design or requirements specification may suggest a programming language. No matter what language is used, each program component involves at least three major aspects: control structures, algorithms, and data structures.

4. Testing

In developing a large system, testing usually involves several stages. First, each program component is tested on its own, isolated from the other components in the system. Such testing, known as module testing, component testing, or unit testing, verifies that the component functions properly with the types of input expected from studying the component's design. Unit testing is done in a controlled environment whenever possible, so the test team can feed a predetermined set of data to the component being tested and observe what output actions and data are produced. In addition, the test team checks the internal structures, logic, and boundary conditions for the input and output data.

When collections of components have unit-tested, the next step is ensuring that the interfaces among the components are defined and handled properly. Integration testing is the process of verifying that the system components work together as described in the system and program design specifications.

Once we are sure that information is passed among components in accordance with the design, we test the system to assure that it has the desired functionality. A function test evaluates the system to determine if the functions described by requirements specification are actually performed by the integrated system.

Recall that the requirements are documented in two ways: first in the customer's terminology and again as a set of the software and hardware requirements the developers could use. The function test compares the system being built with the functions described in the developer's requirements specification. Then, a performance test compares the system with the remainder of these software and hardware requirements. When the test is performed successfully in a customer's actual working environment, it yields a validated system.

When the performance test is complete, we developers are certain that the system functions according to our understanding of the system description. The next step is conferring with the customer to make certain that the system works according to customer expectations. We join the customer to perform an acceptance test, where the system is checked against the customer's requirements description. Upon completion of acceptance testing, the accepted system is installed in the environment in which it will be used; a final installation test is run to make sure that the system still functions as it should.

Figure 9-2 illustrates the relationship among these testing steps. No matter the size of the system being tested, the type of testing described in each step is necessary for assuring proper functioning.

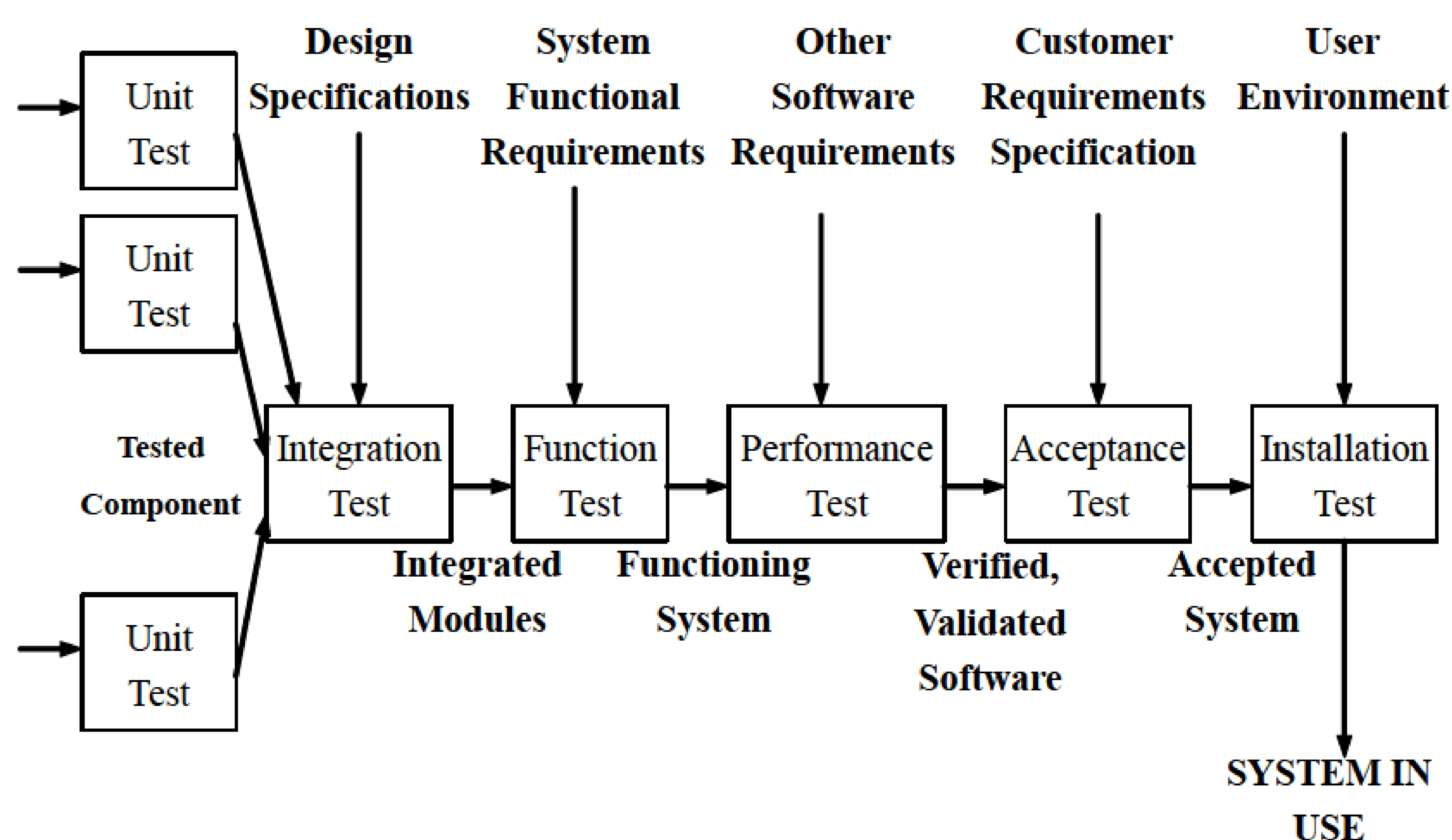


Figure 9-2　Testing Steps

5. Maintenance

System development is complete when the system is operational; that is, when the system is being used by users in an actual production environment. Any work done to change the system after it is in operation is considered to be maintenance. Maintenance activities are similar to those of development: analyzing requirements, evaluating system and program design, writing and reviewing code, testing changes, and updating documentation. So the people who perform maintenance — analysts, programmers, and designers — have similar roles. However, because changes often require an intimate knowledge of the code's structure and content, programmers

play a much larger role in maintenance than they did in development.

Maintenance focuses on four major aspects of system evolution simultaneously:

- Maintaining control over the system's day-to-day functions;
- Maintaining control over system modifications;
- Perfecting existing acceptable functions;
- Preventing system performance from degrading to unacceptable levels.

Maintenance includes the following four types: corrective maintenance, adaptive maintenance, perfective maintenance and preventive maintenance.

9.4 Computer-Aided Software Engineering (CASE)

1. What Is Computer-Aided Software Engineering?

CASE stands for Computer Aided Software Engineering; it can be used to mean any computer-based tool for software planning, development, and evolution.

Computer-aided systems engineering (CASE) is the application of information technology to systems development activities, techniques, and methodologies. CASE tools are programs (software) that automate or support one or more phases of a systems development life cycle. The technology is intended to accelerate the process of developing systems and to improve the quality of the resulting systems. But CASE itself is not a methodology or an alternative to methodologies.

2. CASE Architecture

CASE tools are classified according to which phases of the life cycle they support. The term upper-CASE describes tools that automate or support the "upper" or earliest phases of systems development — the survey, feasibility study, requirements definition, and design phases. The term lower-CASE describes tools that automate or support the "lower" or later phases of systems development — detailed design, construction, and implementation (and also support).

At the center of any true CASE tool's architecture is a database called a repository (see Figure 9-3). A CASE repository is a developers' database. It is a place where the developers can store diagrams, descriptions, specifications, and other by-products of systems development. Synonyms include dictionary and encyclopedia. Many different CASE tools can share information across a single repository.

3. The Benefits of CASE

The overriding goal of CASE technology is the automation of the entire software development life cycle process using a set of integrated software tools, techniques, and methodologies.

Some of the most commonly cited benefits include:

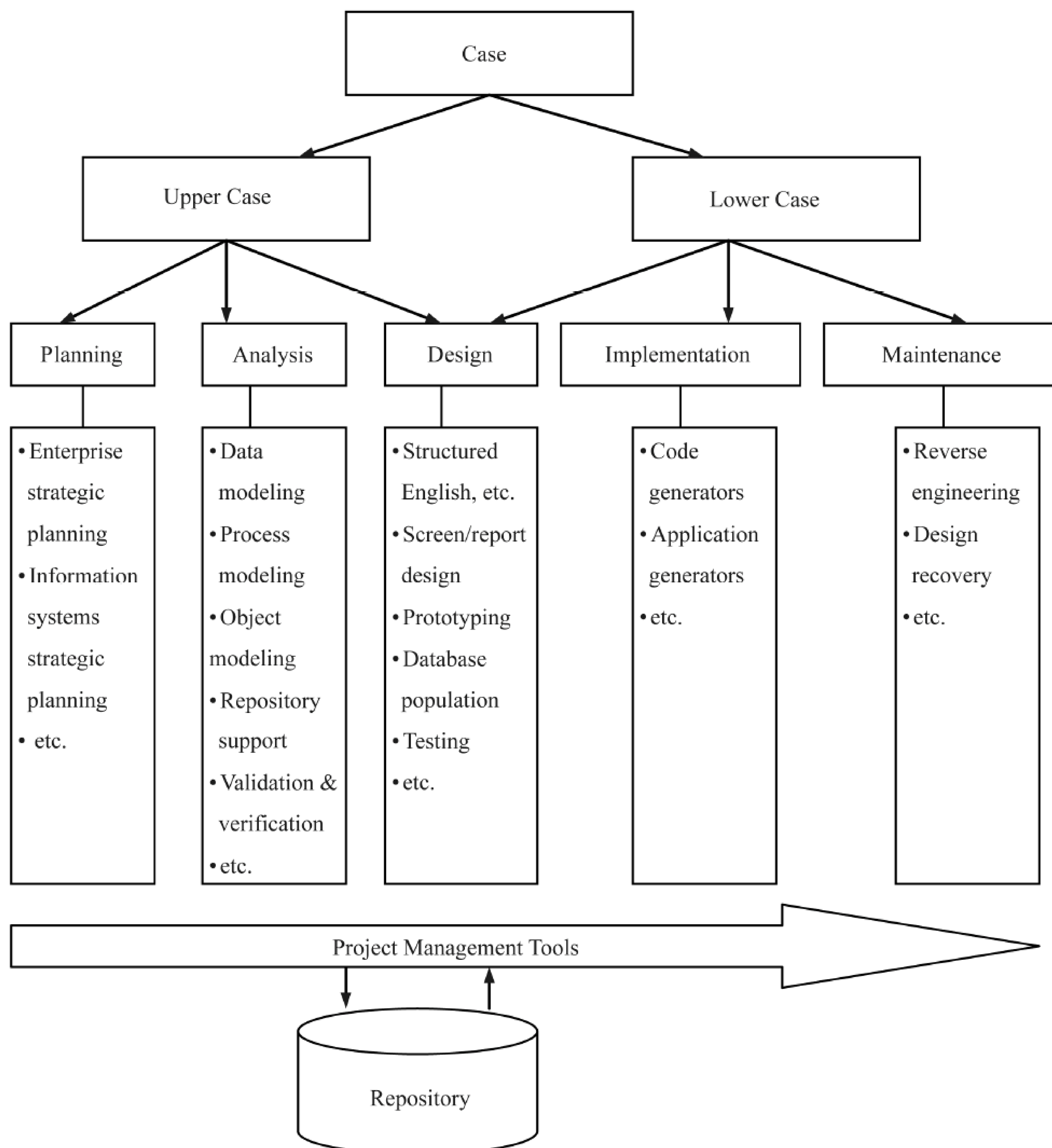


Figure 9-3　CASE Architecture

- Improved productivity (through automation of tasks and rapid application development);
- Improved quality (because CASE tools check for completeness, consistency, and contradictions);
- Better documentation (mostly because the tools make it easier to create and assemble consistent, high quality documentation);
- Reduced lifetime maintenance (because of the aforementioned system quality improvements combined with better documentation);
- Methodologies that really work (through rule enforcement and built-in expertise).

Technical Notes to the Text

1. software engineering，软件工程。软件工程是将系统化的、规范的、定量化的方法用于

软件系统开发、运行和维护，即工程化思想在软件系统开发中的应用。

2. prototyping，原型化。它通过快速地建立一个原型化模型(prototype)，然后再通过与用户反复多次地交互(交流)来逐步完善系统。

3. system architecture，系统体系结构。软件体系结构是具有一定形式的结构化元素，即构件的集合，包括处理构件、数据构件和连接构件。处理构件负责对数据进行加工，数据构件是被加工的信息，连接构件把体系结构的不同部分组合连接起来。

4. reuse，重用。重用是提高软件生产率和质量的一种技术，将已有软件的各种知识用于建立新的软件，以缩减软件开发和维护的费用。

5. software process，软件过程。软件过程是一个将用户需求转化为软件系统所需要的活动的集合，即开发和维护软件及其相关产品所涉及的一系列活动。

6. implementation，实施(或实现)。实施是指构造/组装软件系统技术部件，并最终使软件系统投入运行的过程。

7. unit testing，单元测试。单元测试又称模块测试(module testing)，用于测试单个程序模块，确定模块的逻辑和功能是否正确。

8. module，模块。模块是指具有相对独立性的一组逻辑上有关的实体，其主要成分是一组说明和一组语句。

9. module testing，模块测试。同单元测试。

10. integration testing，集成测试。集成测试用来测试模块之间的接口，即模块之间的数据和控制传递。

11. system testing，系统测试。系统测试是对软件系统中的应用程序、硬件、手工操作，以及系统任何其他组成部分的集成的总体测试。

12. maintenance，维护。计算机软件投入运行后，为了保证系统能够正常工作、进一步满足用户新的需求所采取的对原系统的修改、完善等措施。

13. conceptual design，概念设计，又称总体设计。其主要任务是描述、组织和构造新系统的体系结构，包括软件体系结构设计、信息系统体系结构设计、网络设计、代码设计等内容。

14. technical design，技术设计，又称详细设计。其属于低层设计，包括输入设计、输出设计、界面设计、应用程序设计、数据库设计等。

15. function test，功能测试。功能测试也称有效性测试，它重点测试一个完整的程序所产生的结果是否满足用户的需求和期望。

16. acceptance test，验收测试。验收测试是由用户来完成的测试。

17. corrective maintenance，校正性维护。由于排错不彻底，对软件投入运行后所暴露出来的程序错误进行测试、诊断、定位、纠错及验证修改的回归测试过程称为校正性维护。

18. adaptive maintenance，适应性维护。当系统运行环境(如硬件平台、软件平台)发生变化时，为了使系统适应新的环境进行的维护称为适应性维护。

19. perfective maintenance，完善性维护。完善性维护是系统维护中工作量最大的一部分工作，是指系统投入运行后，根据用户新的需求对系统的功能和质量所做的修改和补充。

20. preventive maintenance，预防性维护。为了使系统在将来具有更好的可靠性和可维护性，事先对软件进行的修改或补充，称为预防性维护。

21. CASE (computer aided software engineering)，计算机辅助软件工程。其指能够支持或使软件开发生命周期中一个或多个阶段自动化的计算机程序(软件)。

22. upper-CASE，上游 CASE，或称前端 CASE(Front-End CASE)。它描述了支持系统开发生命周期(SDLC)前期几个阶段(系统规划、系统分析和前期的系统设计，即系统总体设计)的 CASE 工具。

23. lower-CASE，下游 CASE，或称后端 CASE(Back-End CASE)。它描述了支持系统开发生命周期(SDLC)后期几个阶段(详细系统设计、系统实施和系统维护)的 CASE 工具。

24. CASE repository，CASE 资源库。所谓 CASE 资源库是一个开发者的数据库，该数据库存储了图表、描述、规格说明、应用程序及其他的一些系统开发副产品，也称为设计数据库(design database)、字典(dictionary)或百科全书(encyclopedia)。

25. NATO (North Atlantic Treaty Organization)，北大西洋公约组织，或称北约组织。

26. IEEE (the Institute for Electrical and Electronic Engineers)，美国电气和电子工程师协会。

Word Bank to the Text

A. Useful new words

popularity	*n*. 普及，流行
quantifiable	*adj*. 可以计量的
maintenance	*n*. 维护，保持
daunting	*adj*. 使人畏缩的
abstraction	*n*. 提取
consistency	*n*. 始终一贯，前后一致
primitive	*adj*. 原始的，简单的
functionality	*n*. 函数性，泛函性
iterative	*adj*. 重复的，反复的
embed	*v*. 使插入，使嵌入
explicit	*adj*. 清晰的，明确的
decomposition	*n*. 分解，腐烂
mutually	*adv*. 互相地，互助地
exclusive	*adj*. 排外的，独占的，唯一的
prescriptive	*adj*. 说明性的
configuration	*n*. 构造，结构，配置
rigorous	*adj*. 严格的，严厉的
causal	*adj*. 表示原因的，因果关系的
commonality	*n*. 共同或普通的性质或状态
barrier	*n*. 障碍
specificity	*n*. 特异性，特征
quantitative	*adj*. 数量的，定量的

heterogeneous	*adj*. 异类的，不同的
prescribe	*v*. 指示，规定
elicit	*v*. 得出，引出
verification	*n*. 确认，查证
validation	*n*. 确认
flexibility	*n*. 弹性，适应性，机动性
predetermine	*v*. 预定，预先确定
terminology	*n*. 术语学，专门名词
degrading	*adj*. 丧失体面的，降格的
repository	*n*. 知识库，仓库
overriding	*adj*. 最重要的，高于一切的
aforementioned	*adj*. 上述的，前述的

B. Useful expressions

satisfy users' needs	满足用户需求
tackle the problem	解决问题
concentrate on	集中精力
get mired in	陷入泥潭
settle on	决定
regardless of	不管，不顾
isolate from	使孤立
confer with	商讨，交换意见
in a sense	在某种意义上
in terms of	根据，按照
partition into	分割
capitalize on	利用
pay attention to	注意
contribute to	导致
in particular	特别地
take advantage of	利用
search for	搜寻
easier said than done	说起来容易做起来难
subject to	使服从，使遭受
be composed of	由……组成
agree on	对……达成协议
transform into	转换
move back and forth	前后移动
upon completion of	一完成……就……

in an actual environment	在实际环境中
play a role in	起作用
combine with	与……结合

C. Technical terms and proper names

software engineering	软件工程
prototyping	原型化
system architecture	系统体系结构
reuse	复用，重用
software process	软件过程
requirement	需求
system design	系统设计
program design	程序设计
implementation	实施(或实现)
unit testing	单元测试
module testing	模块测试
component testing	部件测试
integration testing	集成测试
system testing	系统测试
system delivery	系统发布
maintenance	维护
conceptual design	概念设计
technical design	技术设计
function test	功能测试
performance test	性能测试
acceptance test	验收测试
installation test	安装测试
corrective maintenance	校正性维护
adaptive maintenance	适应性维护
perfective maintenance	完善性维护
preventive maintenance	预防性维护
CASE (computer aided software engineering)	计算机辅助软件工程
upper-CASE	上游 CASE
lower-CASE	下游 CASE
CASE repository	CASE 资源库

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. __________ is the application of a systematic, disciplined, quantifiable approach to development, operation, and maintenance of software.

2. An __________ is a description of the problem at some level of generalization that allows us to concentrate on the key aspects of the problem without getting mired in the details.

3. __________ means building a small version of a system, usually with limited functionality that can be used to help the user or customer identify the key requirements of a system and demonstrate feasibility of a design or approach.

4. The software development process is sometimes called the __________, because it describes the life of a software product from its conception to its implementation, delivery, use, and maintenance.

5. A __________ is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The more independent the units are, the more modular the architecture is and the more easily we can design and develop the pieces separately. ( )

2. CASE (computer aided software engineering) tools would enhance software development. ( )

3. The conceptual design allows system builders to understand the actual hardware and software needed to solve the customer's problem. ( )

4. Integration testing verifies that the component functions properly with the types of input expected from studying the component's design. ( )

5. The software requirement is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose. ( )

Ⅲ. Match each of the following terms with the appropriate definition.

system engineering	software process	unit testing	CASE

1. ________ The life of a software product from its conception to its implementation, delivery, use, and maintenance.

2. ________ The application of information technology to systems development activities, techniques, and methodologies.

3. ________ A complete electronic circuit that packages transistors and other electronic

components on a small silicon chip.

4. _______ The testing that verifies the component functions properly with the types of input expected from studying the component's design.

Ⅳ. Describe the stages involved in the software development life cycle (SDLC).

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

predetermined	explicit	consistency
alternative	exclusive	maintenance
elicit	primitive	popularity

1. His ________ among working people remains as strong as ever.
2. There's always a lack of ________ in matters of foreign policy.
3. The window had been replaced last week during routine ________.
4. ________ humans needed to be able to react like this to escape from dangerous animals.
5. She was quite ________ about why she left.
6. The city was criticized for being too _________ and uncompetitive.
7. They promised to make enquiries for us, but several phone calls ________ no further information.
8. You have the ________ of marrying or remaining a bachelor.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "establish" and fill in the sentences with the right word.

Example: However, **establishing** a long-term, effective reuse **program** can be difficult, because there are several barriers.

1. The United Nation has established detailed cr______ for who should be allowed to vote.
2. The students of arts had already established co______ with the museum.
3. China and Saudi Arabia announced they were establishing formal diplomatic re______.
4. An autopsy was being done to establish the ca______ of death.
5. His second novel established his fa______ as a writer.
6. The police have established his in______.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software. It includes techniques and procedures, often __1__ by a software development process, with the purpose __2__ improving the

reliability and maintainability of software systems. The effort is __3__ by the potential complexity of those systems, which may contain millions of __4__ of code.

The term software engineering was popularized by F.L. Bauer __5__ the NATO Software Engineering Conference in 1968. The discipline of software engineering includes knowledge, tools, and methods for software requirements, software design, software construction, software testing, and software __6__ tasks. Software engineering is related __7__ the disciplines of computer science, computer engineering, management, mathematics, project management, quality management, software ergonomics, and systems engineering.

In 2004, the U.S. Bureau of Labor Statistics counted 760840 software engineers __8__ jobs in the U.S.; in the same time period there were some 1.4 million practitioners employed in the U.S. in all other engineering disciplines combined. Due to its relative __9__ as a field of study, formal education in software engineering is often taught as part of a computer science __10__, and as a result most software engineers hold computer science degrees.

1. A. regulated	B. regulation	C. regulating	D. regular
2. A. to	B. of	C. with	D. as
3. A. necessary	B. necessitated	C. necessity	D. necessarily
4. A. draws	B. writings	C. works	D. lines
5. A. on	B. at	C. during	D. in
6. A. maintain	B. maintaining	C. maintenance	D. maintained
7. A. to	B. with	C. for	D. due
8. A. holding	B. having	C. owning	D. finding
9. A. news	B. newness	C. new	D. newly
10. A. task	B. subject	C. engineer	D. curriculum

Translation

Ⅷ. Translate the following into Chinese.

1. Just as manufactures look for ways to assure the quality of the products they produced, so too must software engineers find methods to assure that their products are of acceptable quality and utility.

2. An abstraction is a description of the problem at some level of generalization that allows us to concentrate on the key aspects of the problem without getting mired in the details.

3. The importance of these approaches is their capture of our design experience, enabling us to capitalize on our past projects by reusing both what we have done and what we learned by doing it.

4. Upon completion of acceptance testing, the accepted system is installed in the environment in which it will be used; a final installation test is run to make sure that the system still functions as it should.

5. Software development is both a creative and a step-by-step process, often involving many people producing many different kinds of products.

Chapter 10
Object-Oriented Technology

Pre-reading Questions

1. What is encapsulation? Describe the advantages of encapsulation.
2. What is polymorphism? Describe the advantages of polymorphism.
3. How many graphical diagrams are defined in UML? Describe each of them.

10.1 A Brief Overview of Object Technology

The object-oriented (OO) paradigm is a development strategy based on the concept that systems should be built from a collection of reuseable parts called objects. Examples of OO languages and technologies include the Java, C#, and C++ programming languages and the Enterprise JavaBeans (EJB) framework. The original motivation of the object paradigm was that objects were meant to be abstractions of real-world concepts. This was absolutely true of business objects, but business objects are only part of the picture — you also need user interface objects to enable your users to work with your system, process objects that implement logic that works with several business concepts, systems objects that provide technical features such as security and messaging, and potentially some form of data objects that persists your business objects. Table 10-1 summarizes the strengths and weaknesses of object technology for business system development.

Table 10-1 Evaluating Object Technology

Advantages	Disadvantages
• Enables development of complex software • Wide industry acceptance • Mature, proven technology • Wide range of development languages and tools to choose from • Very easy to find people with object experience	• Significant skill set is required • No single language dominates the landscape (although Java, C#, C++, and arguably Visual Basic are clearly popular and here to stay) • Not all IT professionals, in particular some within the data community, accept it • Technical "impedance mismatch" with structured technologies and RDBs

10.2 What Is OO — System Concepts for Object Modeling

Object modeling is a technique for identifying objects within the system environment and the relationships between those objects. The object-oriented approach to system development is based

on several concepts, such as object, attribute, behavior, encapsulation, class, inheritance, polymorphism, persistence, etc.

1. Objects, Attributes, Methods, and Encapsulation

Webster's dictionaries definition of an object is that "something that is or is capable of being seen, touched, or otherwise sensed." In object-oriented approaches to systems development the definition of an object is as follows: An object is something that is or is capable of been seen, touched, or otherwise sensed, and about which users store data and associate behavior.

Three portions of this definition need to be examined. First, let's consider the term something. That something can be characterized as a type of object. The types of objects may include a person, place, thing, or event. An employee, customer, vendor, and student are examples of person objects. A particular warehouse, regional office, building, and room are examples of place objects. Examples of thing objects include a product, vehicle, equipment, videotape, or a window appearing on user's display monitor. Finally, examples of event objects include an order, payment, invoice, application, registration, and reservation.

Now let's consider the "data" portion of our definition. In object-oriented circles, this part of our definition refers to what are called attributes. Attributes are the data that represent characteristics of interest about an object. For example, we might be interested in the following attributes for the person object customer: CUSTOMER_NUMBER, CUSTOMER_ NAME, TYPE_ OF_CUSTOMER, CREDIT_LIMIT, AVAILABLE_CREDIT, ACCOUNT_ BALANCE, and ACCOUNT STATUS. In reality, there may be many customer objects for which we would be interested in these attributes. Each individual customer is referred to as an object instance. An instance (or object instance) of an object consists of the values for the attributes that describe a specific person, place, thing, or event.

With advances in technology, attributes have involved to include more than simple data characteristics as represented in the previous examples. Today, objects may include newer attribute types, such as a bitmap, a picture, sound, or even videos.

The last portion of our definition for an object is the "behavior" of an object.

Behavior refers to those things that the object can do and that correspond to functions that act on the object's data (or attributes). In object-oriented circles, an object's behavior is commonly referred as a method, operation, or service.

Another important object-oriented principle is that an object is solely responsible for carrying out any functions or behaviors that act on its own data (or attributes). This leads us to an important concept for understanding objects, called encapsulation.

Encapsulation is the packaging of several items together into one unit. Applied to objects, both attributes and behavior of the object are packaged together. The only way to access an object's attributes is through that object's behaviors. No other object may perform that object's behavior.

An OO system uses encapsulation. Encapsulation is the way in which the methods form a

protective boundary around an object, isolating it from things that happen to other objects. That is, objects can be manipulated only through their methods. An object encapsulates an object's behaviors and attributes, hiding the implementation details. However, encapsulation is not the same as information hiding. Encapsulation's protective boundary may in fact be transparent or translucent. That is, sometimes you can see within the protective boundary (i.e., it is transparent and visible), but sometimes it is hidden (i.e., it is translucent). Berard notes that "abstraction is a technique that helps us to identify which specific information should be visible, and which information should be hidden. Encapsulation is then the technique for packaging the information in such a way as to hide what should be hidden, and make visible what is intended to be visible ".

Thus, we can say an object is the encapsulation of its data (or attributes) and behaviors.

2. Classes, Generalizations, and Specialization

An important concept of object modeling is the concept of categorizing objects into classes. That is, OO uses classification to group objects that have attributes and behaviors in common. We say that each object is an instance of a class. Each instance has its own attribute values (that is, it has ways to describe its state at any time) but shares attribute names and behaviors with the other instances of the class. Thus a class describes a set of objects that share a common structure and common behaviors, but attributes values help us distinguish particular objects from one another.

A class is a set of objects that share common attributes and behaviors. A class is sometimes referred to as an object class. For example, student and teacher objects represent thing-objects that have some similar attributes and behaviors and could be classified as persons (see Figure 10-1).

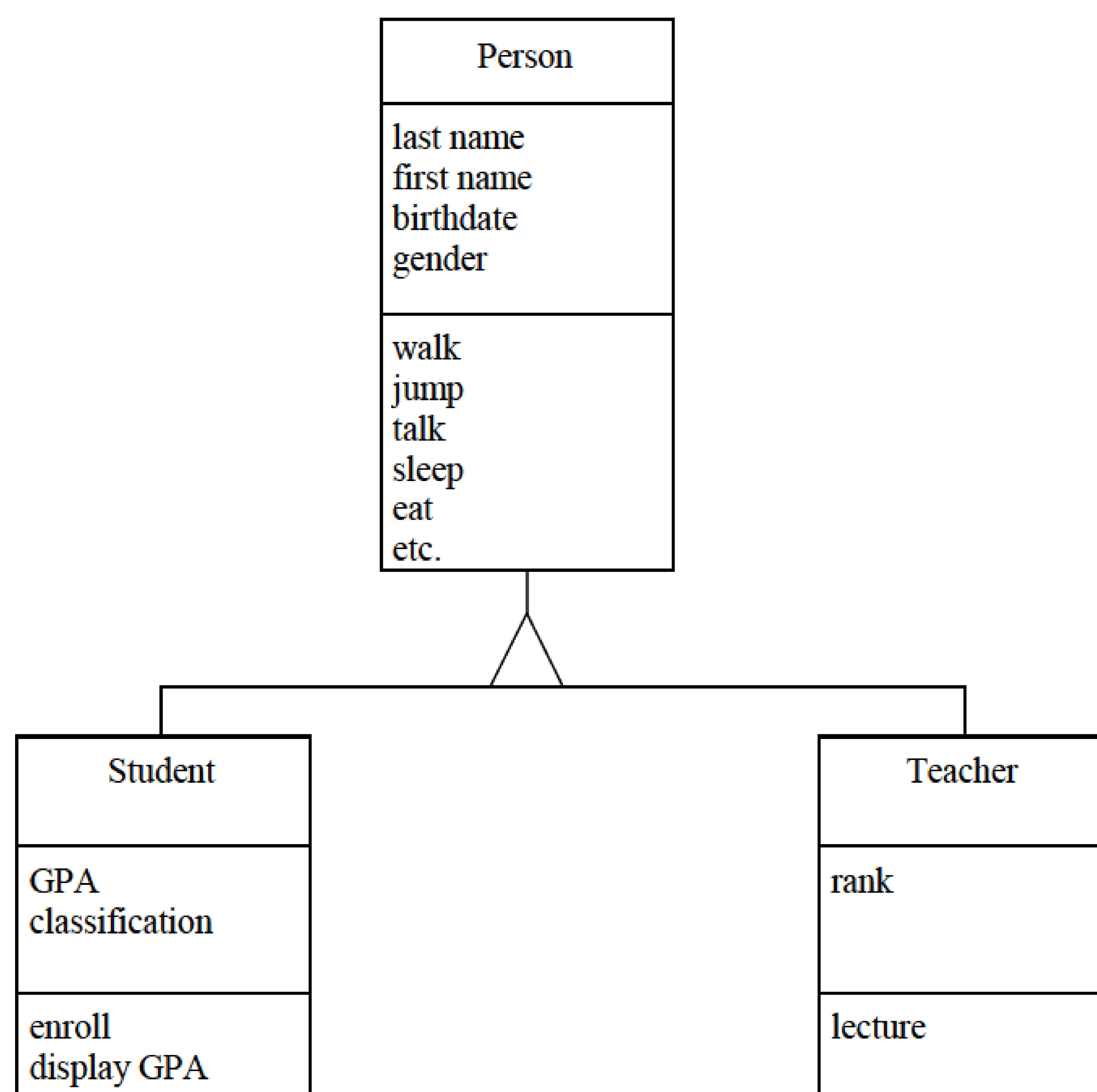


Figure 10-1　Supertype and Subtype Relationships Between Object Classes

A class may also have subclasses of objects. For example, STUDENT and TEACHER object classes could be members of the class PERSON.

When levels of classes are identified, the concept of inheritance is applied. Clearly, some attributes are shared among the members of a given class. We can organize classes hierarchically according to the sameness or differences among them; this hierarchy exhibits the OO classes' inheritance structure. To build the hierarchy, we begin by broadly defining a class and then refining it into more specialized subclasses. A subclass may inherit the structure as well as the behavior and attributes of its superclass. Sometimes we use an abstract class to simplify the hierarchy, where no objects of the abstract class may be defined except as instances of a subclass.

Inheritance means that methods and/or attributes defined in an object class can be inherited or reused by another object class.

The approach that seeks to discover and exploit the commonalties between objects/classes is referred to as generalization/specialization. Generalization/specialization is a technique wherein the attributes and behaviors that are common to several types of an object classes are grouped into their own class, called a supertype. The attributes and methods of the supertype object class are then inherited by those object classes.

In the object class PERSON, STUDENT and TEACHER example, PERSON is referred to as a supertype (or generalization class) whereas STUDENT and TEACHER are referred to as subtypes (or specialization class).

A class supertype is an entity whose instances store attributes that are common to one or more class subtypes. A class subtype is an object class whose instances inherit some common attributes from a class supertype, and then add other attributes that are unique to an instance of the subtype.

The class supertype will have one or more one-to-one relationships to object class subtypes. These relationships are sometimes called "IS A" relationships (or "WAS A", or "COULD BE A").

3. Object/Class Relationships

Conceptually, objects and classes do not exist in isolation. An object/class relationship is a natural business association that exists between one or more objects/classes. There are many relationships between objects or classes.

(1) Association

Consider the class customer and order that may exist in a typical information system and how they interact.

- A CUSTOMER PLACES zero or more ORDERs.
- An ORDER IS PLACED BY one and only one CUSTOMER.

We graphically illustrate the association (relationship) between two classes as a connecting line, as shown in Figure 10-2 (a). A verb phrase describes the relationship. This relationship is called association.

Notice that all relationships are implicitly bi-directional.

Multiplicity defines the minimum and maximum number of occurrences of one object/class for a single occurrence of the related object/class, as shown in Figure 10-2 (b).

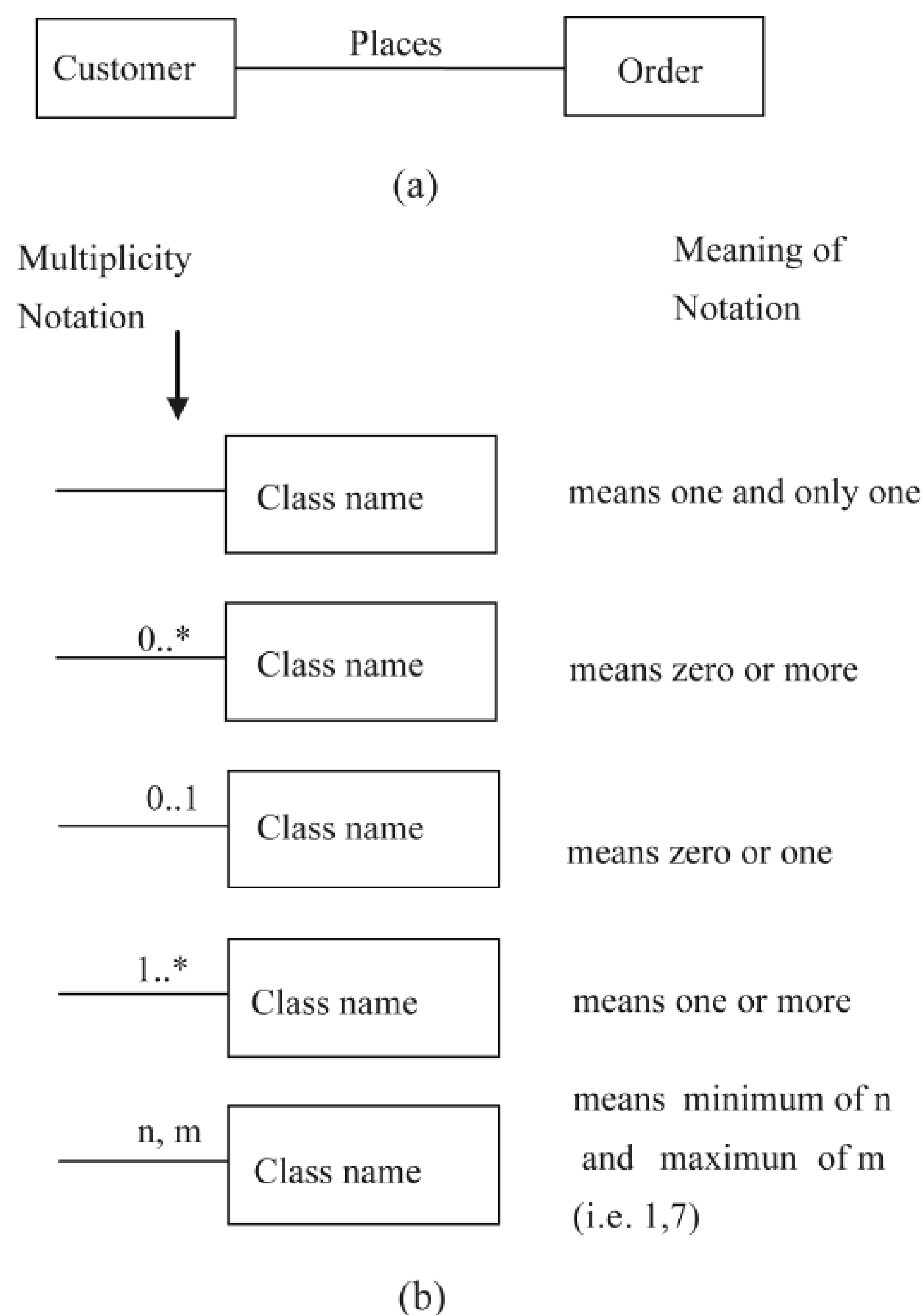


Figure 10-2　Object/Class Associations and Multiplicity Notations

(2) Aggregation

Sometimes objects/classes are made up of other objects/classes. This type of relationship is called aggregation. It is also sometimes referred to as "whole-part" or "part-of" relationships.

For example, the TEXTBOOK object may contain several objects, including: COVER, TABLE OF CONTENTS, CHAPTER, and INDEX objects. The CHAPTER object contains PAGE objects, which in turn contain PARAGRAPH objects, which in turn contain WORD objects, and so forth.

By identifying aggregation relationships we can partition a very complex object and assign behaviors and attributes to the individual objects within it. Notice that multiplicity is also specified for aggregate relationships, as shown in Figure 10-3.

4. Messages and Message Sending

We just learned that objects/classes interact. But how do they interact? Objects/classes interact or "communicate" with one another by passing message. A message is passed when one object invokes another object's method (behavior) to request information or some action (see Figure 10-4). An object sending a message does not need to know how the receiving object is organized internally or how the behavior is to be accomplished, only that it responds to the request in a well-defined way.

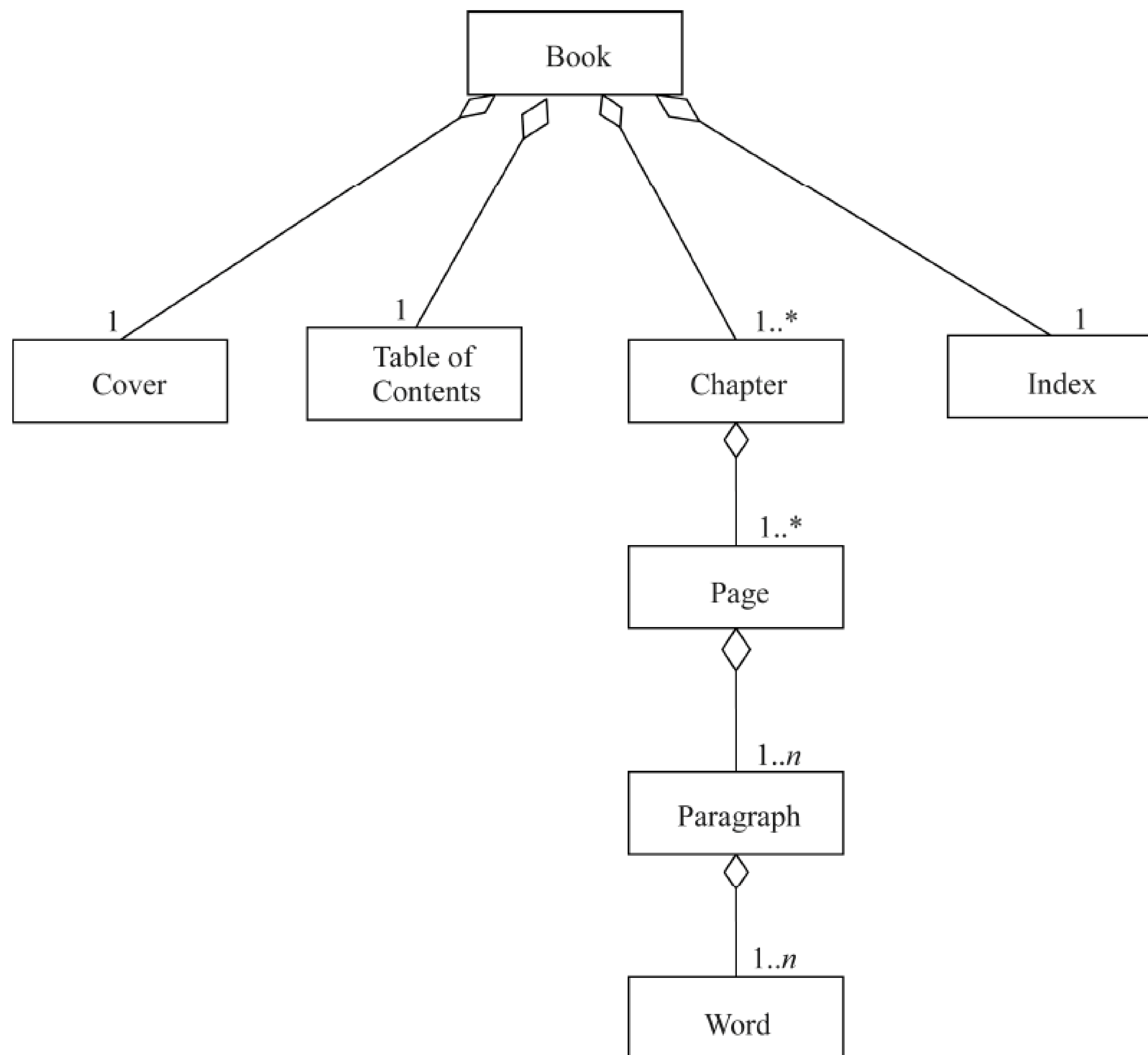


Figure 10-3 Aggregation Relationships

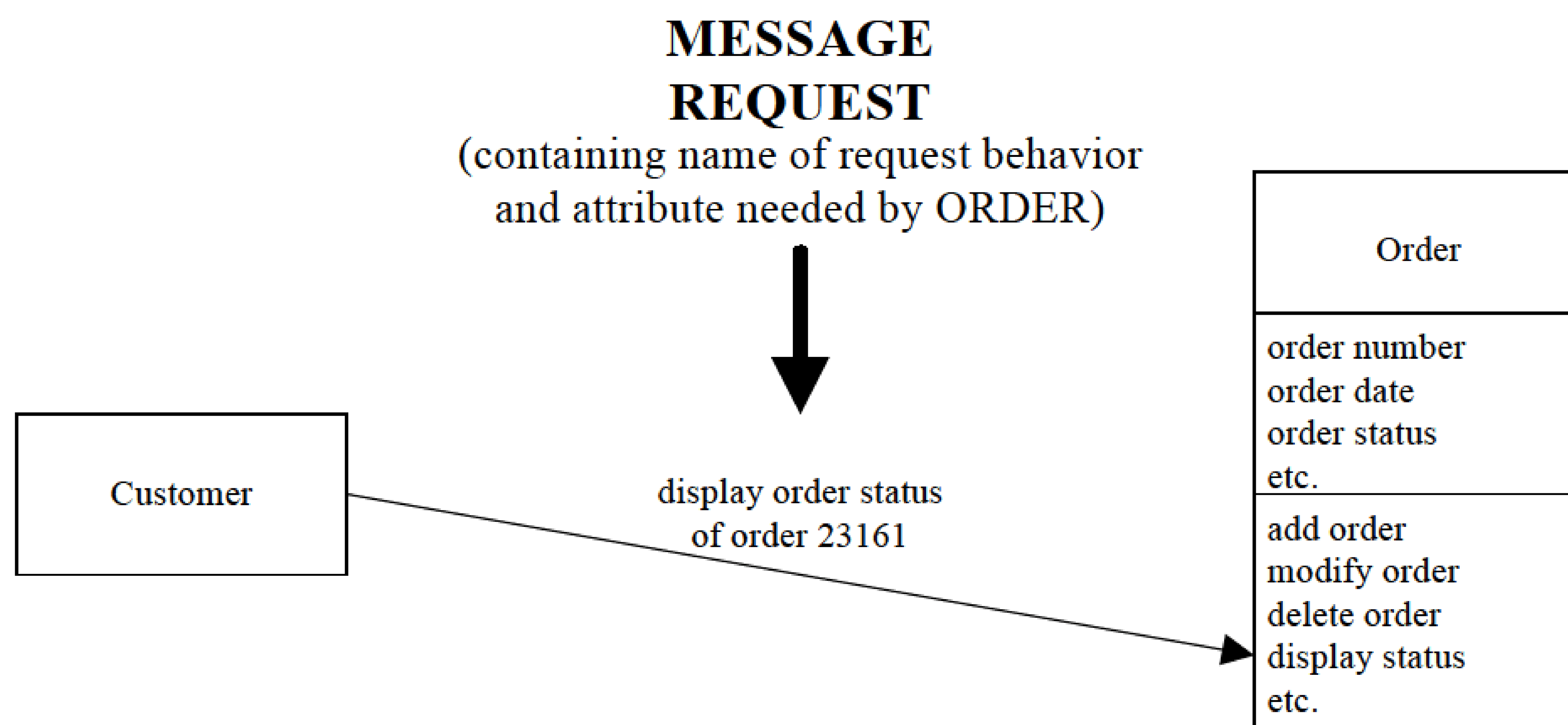


Figure 10-4 Messaging Sending

5. Polymorphism

A behavior is an action that an object performs or to which it is subjected. An object's behavior is triggered by receipt of a particular message, or entrance into a particular state. Sometimes, the same behavior may be exhibited differently on different classes or subclasses, a property called polymorphism.

Polymorphism means "many forms". Applied to object-oriented techniques, it means that a behavior may be completed differently for different objects/classes.

For example, consider the class of polygons. Each polygon has an area, but computing the

area of a triangle is different from computing the area of a rectangle. Thus, the area calculation is particular to the object to which it is applied. An OO programming language is designed to select automatically the correct method to implement an operation, data associated with the operation as parameters, and the name of the object's class. In our area example, the object will select the proper area method, based on the parameters that describe the polygon. Polymorphism allows new classes to be added without changing the existing code.

So how is polymorphism related to message sending? The requesting object knows what service (or behavior) to request and from which object. However, the requesting object does not need to worry about how a behavior is accomplished.

6. Persistence

Another property associated with OO systems is persistence: the ability of an object's name, state and behaviors to transcend time or space. In other words, the object's name, state and behaviors are saved as the object is transformed. For example, we may want our system to save cost information for each gallon of diesel fuel. The price may change each day, so that a person paying $1.5 per gallon today may pay $1.55 tomorrow. But we want the daily price to persist so we can compare last year's price with this year's, or graph the changes over time. In this case, the object persists, even as its attributes are transformed.

10.3　The OO Development Process

One advantage of OO development is its consistency of language. We can describe both the problem and the solution in the same terms: classes, objects, attributes and behaviors. Throughout the development process, we should have consistency of terminology and of perspective. For example, no matter where in the process we represent an object, we must include the object's name, attributes, and behaviors. Describing classes using OO representation requires three perspectives: static, dynamic, and restrictive. The static views include descriptions of the objects, attributes, behaviors, and relationships. The dynamic views describe communication, control/timing, and the states and state changes. The restriction views describe constraints on the structure (such as attribute values or cardinality) and the dynamic behavior.

This across-the-process consistency is a key difference between more traditional procedural development and the OO development process. An OO process uses the encapsulation of data and behavior to form independent units (objects). The same semantic constructs represent the system from the requirements to the application's implementation and testing. Thus, OO is a philosophy of problem and solution representation, not a software life cycle by itself. Indeed, OO can be used in many different software life cycles, from waterfall to spiral. And because OO takes advantage of encapsulation and information hiding, many OO developers like to think of objects and classes in terms of their likelihood for reuse. In general, OO deals with requirements, high-level design, low-level design, coding, and testing, but not necessarily in a sequential fashion; the sequence is

determined by the life cycle, not by the OO representation.

Table 10-2 lists various characteristics of a software product or project. The right-hand column suggests that some characteristics are not likely to change, while others have a high probability of changing substantially during development. The items in this table can help you to decide which development process is most appropriate for the situation at hand.

Table 10-2 Tendency for Change When Using OO Paradigm (Jacobson, 1995)

Characteristic of Software Product/Project	Probability for Change
Objects derived from the application	Low
Long-lived information structures	Low
Passive object's attribute	Medium
Sequences of behavior	Medium
Interface with the outside world	High
Functionality	High

1. OOA

No matter what the life cycle is, an OO development process requires steps for describing requirements, designing the system, designing the programs, coding, and testing. OO requirements analysis is usually done in the user's language and discusses the concepts and scenarios likely in the application domain. The concepts include information, services, and responsibilities. Domain knowledge enables the developers to understand the context in which the system will be used and to describe the requirements in a way that the user will understand them.

This expression of the requirement is the same, no matter how the developers decide to implement the system; that is, the requirements definition can be independent of its representation as objects.

Object-oriented analysis (OOA) is concerned with developing software engineering requirements and specifications that express as a system's object model (which is composed of a population of interacting objects), as opposed to the traditional data or functional views of systems. OOA can yield the following benefits: maintainability through simplified mapping to the real world, which provides for less analysis effort, less complexity in system design, and easier verification by the user, reusability of the analysis artifacts which saves time and costs; and depending on the analysis method and programming language, productivity gains through direct mapping to features of object-oriented programming languages.

An object is a representation of a real-life entity or abstraction. OOA specifies the structure and the behavior of the object — these comprise the requirements of that object. Different types of models are required to specify the requirements of the objects. The information or object model contains the definition of objects in the system, which includes: the object name, the object attributes, and object relationships to other objects. The behavior or state model describes the behavior of the objects in terms of the states the objects exist in, the transitions allowed between objects, and the events that cause objects to change states. These models can be created and

maintained using CASE tools that support representation of objects and object behavior.

OOA views the world as objects with data structures and behaviors and events that trigger operations, or object behavior changes, that change the state of objects. The idea that a system can be viewed as a population of interacting objects, each of which is an atomic bundle of data and functionality, is the foundation of object technology and provides an attractive alternative for the development of complex systems. This is a radical departure from prior methods of requirements specification, such as functional decomposition and structured analysis and design.

Numerous OOA methods have been described since 1988. These OOA methods include: Shlaer-Mellor, Jacobson, Coad-Yourdon, and Rumbaugh. The results of implementing these methods range from tremendous successes at AT&T Bell Labs to a mixture of successes and partial failures on other projects. AT&T Bell Labs realized benefits from OOA on a large project called the Call Attempt Data Collection System (CADCS). Additionally, they found during the development of two releases of the CADCS that use of the OOA techniques resulted in an 8% reduction in requirements specification time and a 30% reduction in requirements staff effort. Other OOA efforts have not been able to reproduce these successes for reasons such as the lack of completed pilot projects, and the lack of formal OOA training.

2. OOD

Normally, the requirements specification (that is, the document intended for use by designers) would be represented in an object-oriented way, so that the designers can generate their designs from the problem description. However, in many cases, the OO problem description is the same as or similar to the first steps of finding an OO solution. So during OO development, the requirements specification steps may actually be the first steps of system design. That is, in both cases, the objects must be identified and their interrelationships described.

There are two guidelines that apply to representing a system design in an object-oriented way. First, it is important to identify and represent classes and objects. We must know not only real-world (problem domain) objects but also the details of each object's attributes and behaviors. Second, we must identify the interactions and relationships among objects and classes: their associations, compositions, aggregations, and inheritance relationships.

The system design is considered to be a high-level abstraction of what will eventually be the program design. Beginning with the system design, the program designers take several steps to provide the details essential for implementation:

- They insert computational features in the models.
- They insert some class library details, usually using a bottom-up approach.
- They consider nonfunctional requirements, such as performance and security, and enhance the design accordingly.

Object-oriented design (OOD) is concerned with developing an object-oriented model of a software system to implement the identified requirements. Many OOD methods have been described since the late 1980s. The most popular OOD methods include Booch, Buhr, Wasserman,

and the HOOD method developed by the European Space Agency. OOD can yield the following benefits: maintainability, reusability and higher productivity.

OOD builds on the products developed during OOA by refining candidate objects into classes, defining message protocols for all objects, defining data structures and procedures, and mapping these into an object-oriented programming language (OOPL). Several OOD methods (Booch, Shlaer-Mellor, Buhr, Rumbaugh) describe these operations on objects, although none is an accepted industry standard. Analysis and design are closer to each other in the object-oriented approach than in structured analysis and design. For this reason, similar notations are often used during analysis and the early stages of design. However, OOD requires the specification of concepts nonexistent in analysis, such as the types of the attributes of a class, or the logic of its methods.

Design can be thought of in two phases. The first, called high-level design, deals with the decomposition of the system into large, complex objects. The second phase is called low-level design. In this phase, attributes and methods are specified at the level of individual objects. This is also where a project can realize most of the reuse of object-oriented products, since it is possible to guide the design so that lower-level objects correspond exactly to those in existing object libraries or to develop objects with reuse potential. As in OOA, the OOD artifacts are represented using CASE tools with object-oriented terminology.

3. OOPL

Once the program design is done, the system is described at a very low level, using models of objects, attributes, and behaviors. Coding proceeds by translating the models to an OO programming language. This step is not perfunctory. It is usually necessary for the implementers to refine the hierarchical structures and make adjustments as the requirements grow and mature. In some situations, the implementers also look for opportunities to make some of the system more general, so that the objects and classes are more suitable for reuse in the future.

Object-oriented programming languages (OOPLs) are the natural choice for implementation of an Object-Oriented Design because they directly support the object notions of classes, inheritance, information hiding, and dynamic binding. Because they support these object notions, OOPLs make an object-oriented design easier to implement. An object-oriented system programmed with an OOPL results in less complexity in the system design and implementation, which can lead to an increase in maintainability. The genesis of this technology dates back to the early 1960s with the work of Nygaard and Dahl in the development of the first object-oriented language called Simula 67. Research progressed through the 1970s with the development of Smalltalk at Xerox. Current OOPLs include C++, Objective C, Smalltalk, Eiffel, Common LISP Object System (CLOS), Object Pascal, Java, and Ada 95.

Object-oriented (OO) applications can be written in either conventional languages or OOPLs, but they are much easier to write in languages especially designed for OO programming. OO language experts divide OOPLs into two categories: hybrid languages and pure OO languages.

Hybrid languages are based on some non-OO model that has been enhanced with OO concepts. C++, Ada 95, and CLOS (an object-enhanced version of LISP) are hybrid languages. Pure OO languages are based entirely on OO principles; Smalltalk, Eiffel, Java, and Simula are pure OO languages.

In terms of numbers of applications, the most popular OO language in use is C++. One advantage of C++ for commercial use is its syntactical familiarity to C, which many programmers already know and use; this lowers training costs. Additionally, C++ implements all the concepts of object orientation, which include classes, inheritance, information hiding, polymorphism, and dynamic binding. One disadvantage of C++ is that it lacks the level of polymorphism and dynamics most OO programmers expect. Ada 95 is a reliable, standardized language well-suited for developing large, complex systems that are reliable.

The major alternative to C++ or Ada 95 is Smalltalk. Its advantages are its consistency and flexibility. Its disadvantages are its unfamiliarity (causing an added training cost for developers), and its inability to work with existing systems (a major benefit of C++).

OOPLs are strongly recommended to complete the implementation of Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD) technologies. AT&T Bell Labs used OOD and OOPLs and realized the benefits of reduced product development time and increased reuse of both code and analysis/design artifacts on a large project called Call Attempt Data Collection System (CADCS). This large project consisted of over 350,000 lines of C++ code that ran on a central processor with over 100 remote systems distributed across the United States. During the development of two releases of the CADCS, the use of the OOD techniques and subsequent implementation in OOPL resulted in a 30% reduction in development time and a 20% reduction in development staff effort as compared to similarly-sized projects using traditional software development techniques and languages.

For applications where OOPL code is to be generated by a CASE tool, developers must decide which programming language to generate: C++, Ada 95, Smalltalk, Java, or CLOS. The choice of an OOPL can limit the choices of CASE tools because the tools may not support the chosen language. However, if language generation is not a consideration, then CASE tools can be chosen based on features and design capabilities without regard to the OOPL chosen for implementation.

Since different OOPLs support different levels of “objectiveness” (e.g., inheritance), different OOD constructs may or may not map directly to OOPL constructs. Therefore, the choice of an OOPL is affected by a design captured using OOD techniques. Where OOD is not present, any OOPL can be used, depending upon the training of the developers.

10.4 Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML offers a standard way to write a system’s blueprints,

including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components. The UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a powerful notation and a standard language for building software blueprints. Three prominent object-oriented programming professionals, Gray Booch, Ivar Jacobsen, and James Rumbaugh are the principle authors of UML. UML establishes a collection of graphical symbols as well as semantics to support and define these symbols. This collection can be broken down into three kinds of building blocks: things, relationships, and diagrams. Things are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams group interesting collections of things.

The choice of what models and diagrams one creates has a profound influence upon how a problem is attacked and how a corresponding solution is shaped. Abstraction, the focus on relevant details while ignoring others, is a key to learning and communicating for the following three reasons:

- Every complex system is best approached through a small set of nearly independent views of a model. No single view is sufficient.
- Every model may be expressed at different levels of fidelity.
- The best models are connected to reality.

In terms of the views of a model, the UML defines the following graphical diagrams:

- Use case diagram. A use case is a behaviorally related sequence of steps (a scenario), both automated and manual for the purpose of completing a single business task. Use case diagrams graphically depict the interactions between the system and the external systems and users. In other words, they graphically describe who will use the system and in what ways the user expects to interact with the system.
- Class diagram. Class diagrams depict the system's object structure. They show object classes that the system is composed of as well as the relationships between those object classes.
- Statechart diagram. Statechart diagrams represent the behavior of entities capable of dynamic behavior by specifying its response to the receipt of event instances. Typically, it is used for describing the behavior of classes, but statecharts may also describe the behavior of other model entities such as usecases, actors, subsystems, operations, or methods.
- Activity diagram. An activity diagram is a special case of a state diagram in which all (or at least most) of the states are action or subactivity states and in which all (or at least most) of the transitions are triggered by completion of the actions or subactivities in the source states. The entire activity diagram is attached (through the model) to a class, such as a use case, or to a package, or to the implementation of an operation. The purpose of this diagram is to focus on flows driven by internal processing (as opposed to external events). Use activity diagrams in situations where all or most of the events represent the

completion of internally-generated actions (that is, procedural flow of control). Use ordinary state diagrams in situations where asynchronous events occur.

- Interaction diagrams. A pattern of interaction among instances is shown on an interaction diagram. Interaction diagrams come in two forms based on the same underlying information, specified by an interaction, but each form emphasizing a particular aspect of it. The two forms are: sequence diagrams and collaboration diagrams. Sequence diagrams show the explicit sequence of stimuli and are better for real-time specifications and for complex scenarios. Collaboration diagrams show the relationships among instances and are better for understanding all of the effects on a given instance and for procedural design.
- Sequence diagram. A sequence diagram shows an interaction arranged in time sequence. In particular, it shows the instances participating in the interaction by their "lifelines" and the stimuli they exchange arranged in time sequence. It does not show the associations among the objects.
- Collaboration diagram. A collaboration diagram shows an interaction organized around the roles in the interaction and their links to each other. Unlike a sequence diagram, a collaboration diagram shows the relationships among the objects playing the different roles.
- Component diagram. A component diagram shows the dependencies among software components, including source code components, binary code components, and executable components. For a business, "software" components are taken in the broad sense to include business procedures and documents. A software module may be represented as a component stereotype.
- Deployment diagram. Deployment diagrams show the configuration of run-time processing elements and the software components, processes, and objects that live on them.

Since UML is built upon many of the principles of object-oriented programming including object-oriented analysis, it is likely that UML is also effective at representing real-world entities. One limitation to building UML diagrams that are not going to be directly used to build a software system is that many definitions in UML include references to software systems. Some degree of interpretation is therefore necessary in order to use UML outside its intended purpose.

Technical Notes to the Text

1. Enterprise JavaBean (EJB)，Sun 的 JavaEE 服务器端组件模型。该组件模型规范定义了开发和部署基于事务性、分布式对象应用程序的服务器端软件组件的体系结构。企业组织可以构建它们自己的组件，或从第三方供应商购买组件。这些服务器端组件称作 Enterprise Bean，它们是 Enterprise JavaBean 容器中驻留的分布式对象，为分布在网络中的客户机提供远程服务。Enterprise JavaBean (EJB)是 Sun Microsystems 对 CORBA 的可移植性和复杂性的解决方案。

2. object modeling，对象建模。其是基于对象、属性、行为、封装、类、继承、多态、

持久性等概念来识别系统环境内的对象及这些对象之间联系的技术。

3. attribute，属性。对象或类的属性描述了对象的具体特征。属性包括属性名和属性值(或称属性状态)。

4. method，方法。方法也称为服务(service)或操作(operation)，是软件系统为满足用户需求必须采取的行动，是软件系统对事件的响应。事件是某一时刻所发生的事情。

5. generalization/specialization，泛化/特化。其是将拥有共同属性和行为的几种类型的对象类组成一个称为超类型的技术，然后超类型的属性和方法被这些对象类继承。

6. supertype，超类。在类的继承关系中，位于上层的类称为超类。subtype，子类。在类的继承关系中，位于下层的类称为子类。

7. association，关联。关联反映了类和类之间的静态关系。实例连接(instance connection)或链(link)是关联的实例化，反映了两个具体对象之间的关系。

8. multiplicity，多重性。多重性指定义了与一个对象/类相联系的对象/类出现一次时，该对象/类可能出现的最小和最大的数目。

9. aggregation，聚集。有些对象/类是由其他对象/类组合而成的，这种类型的联系称为聚集或整体—部分联系。

10. polymorphism，多态性。多态意味着“多种形式”。在面向对象技术中，多态是指一种行为可以被不同的对象/类以不同的方式执行。

11. persistence，持久性。持久性指对象的生存期可以超越程序的执行实践而长期存在。

12. dynamic binding，动态绑定(联编)。动态绑定指在运行时根据对象接收的消息动态地确定要连接哪一段服务代码。

13. waterfall，瀑布模型。该模型是一种软件系统或信息系统开发生命周期模型，在该模型中各阶段的工作基本按顺序的方式进行，不允许返工。

14. spiral，螺旋模型。该模型综合了瀑布模型和原型开发模型的优点，同时增加了一个新的元素，即风险分析(risk analysis)，用来弥补两者的不足。该模型适合于大型软件系统的开发。

15. maintainability，可维护性。可维护性指修改软件系统或部件以校正错误、提高性能或适应环境变化的容易程度。

16. reusability，可重用性。可重用性指软件模块或其他产品能够被一个以上的计算机程序或软件系统重复使用的程度。

17. unified modeling language (UML)，统一建模语言。

18. use case diagram，用例图。用例图是指反映活动者、系统边界所封闭的用例，以及活动者与用例之间、用例与用例之间关系的一种图。用例图用于描述系统的环境和系统的功能需求，对系统的动态行为建模。

19. class diagram，类图。类图表达了系统中类与类之间的相互关系，这些关系包括关联、泛化和聚集及各种依赖关系。类图属于对象模型中的静态视图。

20. statechart diagram，状态图。状态图可用于描述用户接口、设备控制器和其他具有反馈的子系统，还可用于描述在生命期中跨越多个不同性质阶段的被动对象的行为，在每一阶段该对象都有自己特殊的行为。

21. activity diagram，活动图。活动图是状态图的一个特例，其中所有的状态都是动作状

态，且转换都是由源状态中动作的完成触发的。活动图与特定的类或用例相关，它描述某个方法的内部表现。使用活动图可以表示由内部生成的动作驱动的事件流。

22. interaction diagram，交互图。交互图用来描述对象之间及对象与参与者之间的动态协作关系，以及协作过程中行为次序的图形文档。

23. sequence diagram，顺序图。顺序图表示了对象之间传送消息的时间顺序，也就是对象之间的交互顺序，这些交互是指在场景或用例的事件流中发生的。

24. collaboration diagram，协作图。协作图对在一次交互中有意义的对象和对象间的链建模。

25. component diagram，构件(组件)图。构件图表达了构件及构件间的接口和依赖关系。

26. deployment diagram，部署(配置)图。部署图是显示运行时系统结构的实现级别的图表。通过部署图，可以了解构成应用程序的硬件和软件元素的配置和部署方式。部署图由结点、构件及它们之间的关系构成。结点是一种计算资源，即运行时的物理对象。作为最低要求，结点要有内存和某种计算能力。

Word Bank to the Text

A. Useful new words

encapsulation	*n*. 包装，封装
polymorphism	*n*. 多形性，多态现象
inheritance	*n*. 遗传，遗产
attribute	*n*. 属性，品质，特征
translucent	*adj*. 半透明的
transparent	*adj*. 透明的，透光的
occurrence	*n*. 发生，出现
polygon	*n*. 多角形，多边形
transcend	*v*. 超越，胜过
static	*adj*. 静止的，静态的
dynamic	*adj*. 动力的，动态的
constraint	*n*. 约束，限制
spiral	*adj*. 螺旋形的
substantially	*adv*. 实质上地，相当大地
maintainability	*n*. 可维护性
radical	*adj*. 根本的，基本的
prior	*adj*. 优先的，在前的
numerous	*adj*. 众多的，许多的
eventually	*adv*. 最后，终于
accordingly	*adv*. 因此，从而
artifact	*n*. 人造物品，制品

genesis	*n*. 起源
syntactical	*adj*. 依照句法的
familiarity	*n*. 熟悉，通晓
prominent	*adj*. 卓越的，显著的
fidelity	*n*. 忠实，诚实
asynchronous	*adj*. 不同时的，[电]异步的
explicit	*adj*. 明确的，清楚的
stimulus	*n*. 刺激物，促进因素
collaboration	*n*. 协作

B. Useful expressions

be capable of	能够
in reality	实际上，事实上
with advances in	随着……的进步
correspond to	相应，符合
carry out	贯彻，执行
categorize into	归类，分门别类
have sth. in common	具有共同之处
be classified as	分类
refine into	精炼成
define as	下定义
group into	组成
be subjected to	使经历，使遭受
associate with	联系，联合
take advantage of	利用
be likely to	有可能
be appropriate for	适于，合乎
at hand	在手边，在附近，即将到来
be composed of	由……组成
an alternative for	对……替选
benefit from	受益
lack of	缺乏
be considered to be	被认为是
date back to	回溯到，远在……(年代)
without regard to	不考虑，不遵守
break down into	划分为
have an influence upon	对……影响
be attached to	连在……上，附属于
in time sequence	按时间的次序

in the broad sense	从广义上讲
participate in	参加

C. Technical terms and proper names

object	对象
class	类
attribute	属性
method	方法
behavior	行为
operation	操作
service	服务
encapsulation	封装
information hiding	信息隐藏
generalization/specialization	泛化/特化
instance	实例
supertype	超类
subtype	子类
association	关联
multiplicity	多重性
aggregation	聚集
persistence	持久性
OOA	面向对象分析
OOD	面向对象设计
OOP	面向对象程序设计
dynamic binding	动态绑定(也称动态联编)
structured analysis and design	结构化分析和设计
unified modeling language (UML)	统一建模语言
use case diagram	用例图
class diagram	类图
statechart diagram	状态图
activity diagram	活动图
interaction diagrams	交互图
sequence diagram	顺序图
collaboration diagram	协作图
component diagram	组件图，构件图
deployment diagram	部署图，配置图

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. An object is the ____________ of its data (or attributes) and behaviors.

2. In an object, both attributes and behavior of the object are packaged together. The only way to access an object's attributes is through that object's ____________.

3. ____________ allows new classes to be added without changing the existing code.

4. Aggregation is also sometimes referred to as ____________ or "part-of" relationships.

5. Generalization/specialization is a technique wherein the attributes and behaviors that are common to several types of an object classes are grouped into their own class, called a ____________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The relationship between the class car and the class wheel is inheritance. (   )

2. The statement that best describes an aggregation relationship is "has-a" relationship. (   )

3. Polymorphism is when an operation in the same class has the same name and a different signature. (   )

4. Three prominent object-oriented programming professionals, Coad, Ivar Jacobsen, and James Rumbaugh are the principle authors of UML. (   )

5. Persistence is the ability of an object's name, state and behaviors to transcend time or space. (   )

Ⅲ. Match each of the following terms with the appropriate definition.

object	encapsulation	class
multiplicity	polymorphism	

1. __________ A set of objects that share common attributes and behavior.

2. __________ Something that is or is capable of being seen, touched, or otherwise sensed, and about which users store data and associate behavior.

3. __________ The packaging of several items together into one unit.

4. __________ The minimum and maximum number of occurrences of one object/class for a single occurrence of the related object/class.

5. __________ A behavior may be completed differently for different objects/classes.

Ⅳ. List two types of relationship between objects/classes.

1. ______________________________

2. ______________________________

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

transcend	transparent	prominent
radical	occurrence	explicit
static	inheritance	numerous

1. The disabled girl feared losing her ________ to her stepmother.
2. The company has to make its accounts and operations as ________ as possible.
3. Divorce has become an everyday ________ in the United States.
4. The housing prices, which have been ________ for several months, are now rising again.
5. The outstanding college student far ________ the others in beauty and intelligence.
6. Despite ________ attempts to diet, her weight soared.
7. A national newspaper carried ________ photographs of himself and his wife.
8. The Football League has announced its proposals for a ________ reform of the way football is run in England.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "attribute…to…" and fill in the sentences with the right word.

Example: Some vendors were discouraged from entering the object modeling area and **attributed** their **discouragement to** the need to support many similar, but slightly different, modeling languages.

1. The di______ was attributed to the carelessness of the engineers, especially their failure to install the pipes regularly.
2. She attributed her broken ma______ to her husband's short temper.
3. Sometimes they attribute their students' poor co______ to a lack of intelligence.
4. Nowadays fewer people accept the idea that women tend to attribute their su______ to external causes such as luck.
5. Some scientists attribute the poor fu______ of the brain to lack of physical exercise.
6. Psychiatrists often attribute juvenile de______ to social and family irresponsibility.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

As hardware and software became __1__ complex, researchers studied ways __2__ which software quality could be maintained. Object-oriented programming was deployed __3__ as an attempt to address this problem __4__ strongly emphasizing discrete units of programming logic and reusability in software.

The Simula programming language was the first to introduce the concepts __5__ object-oriented

programming as a superset of Algol. Simula was used for physical modeling, such as models to study and __6__ the movement of ships and their content through cargo ports. Smalltalk was the first programming language to be __7__ "object-oriented".

Object-oriented programming may be seen as a collection of cooperating objects __8__ opposed to a traditional view in which a program may be seen as a group of tasks to compute. In OOP, each object is __9__ of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an independent little machine with a distinct role or responsibility. The actions or "operators" on the objects are __10__ associated with the object.

1. A. increase	B. increasingly	C. increasing	D. increased
2. A. in	B. for	C. with	D. within
3. A. mainly	B. as a whole	C. in part	D. separately
4. A. over	B. in	C. without	D. by
5. A. underlie	B. underlay	C. underlying	D. underlies
6. A. improve	B. to improve	C. improving	D. improves
7. A. called	B. call	C. to call	D. calls
8. A. as it	B. likes	C. while	D. as
9. A. able	B. made	C. capable	D. thought
10. A. closely	B. close	C. closed	D. closing

Translation

Ⅷ. Translate the following into Chinese.

1. Encapsulation is the way in which the methods form a protective boundary around an object, isolating it from things that happen to other objects.

2. This across-the-process consistency is a key difference between more traditional procedural development and the OO development process.

3. Domain knowledge enables the developers to understand the context in which the system will be used and to describe the requirements in a way that the user will understand them.

4. The system design is considered to be a high-level abstraction of what will eventually be the program design.

5. It is usually necessary for the implementers to refine the hierarchical structures and make adjustments as the requirements grow and mature.

Chapter 11
Introduction to Computer Networks

Pre-reading Questions

1. What is a computer networks?
2. How many types of computer networks do you know?
3. What are the differences between OSI and TCP/IP?

Each of the past three centuries has been dominated by a single technology. The 18th century was the time of the great mechanical systems accompanying the Industrial Revolution. The 19th century was the age of the steam engine. During the 20th century, the key technology has been information gathering, processing, and distribution.

11.1 Data Communications

11.1.1 Signals

1s and 0s cannot be sent as such across network links. They must be further converted into a form that transmission media can accept. Transmission media work by conducting energy along a physical path. So, a data stream of 1s and 0s must be turned into energy in the form of electromagnetic signals.

1. Analog and Digital

Both data and the signals that represent them can take either analog or digital form. Analog refers to something that is continuous — a set of specific points of data and all possible points between. Digital refers to something that is discrete.

Information can be analog or digital. Analog information is continuous. Digital information is discrete.

Signals can be analog or digital. Analog signals can have any value in a range; digital signals can have only a limited number of values.

2. Frequency, Spectrum, and Bandwidth

The electromagnetic signal is a function of time, but it can also be expressed as a function of frequency; that is, the signal consists of components of different frequencies.

The spectrum of a signal is the range of frequencies that is contained. The absolute bandwidth

of a signal is the width of the spectrum. Many signals have an infinite bandwidth. However, most of the energy in the signal is contained in a relatively narrow band of frequencies. This band is referred to as the effective bandwidth, or just bandwidth.

3. Bit Rate and Baud Rate

Two terms used frequently in data communication are bit rate and baud rate. Bit rate is the number of bits transmitted during one second. Baud rate refers to the number of signal units per second that are required to represent those bits.

Computer communication is the process of sharing data, programs, and information between two or more computers. There are five basic types of data communication system.

Off-line data transmission is simply the use of a telephone or similar link to transmit data without involving a computer system. The equipment used at both ends of such a link is not part of a computer, or at least does not immediately make the data available for computer process; that is, the data when sent and/or received are "off-line". This type of data communication is relatively cheap and simple.

Remote batch is the term used for the way in which data communication technology is used geographically to separate the input and/or output of data from the computer on which they are processed in batch mode.

On-line data collection is the method of using communications technology to provide input data to a computer such as input arises — the data are then stored in the computer (say on a magnetic disk) and processed either at predetermined intervals or as required.

Enquiry-response systems provide, as the term suggests, the facility for a user to extract information from a computer. The enquiry facility is passive; that is, it does not modify the information stored. The interrogation may be simple, for example, "RETRIEVE THE RECORD FOR EMPLOYEE NUMBER 1234" or complex. Such systems may use terminals producing hard copy and/or visual displays.

Real-time systems are those in which information is made available to and processed by a computer system in a dynamic manner so that either the computer may cause action to be taken to influence events as they occur (for example as in a process control application) or human operators may be influenced by the accurate and up-to-date information stored in the computer, for example as in reservation systems.

11.1.2 Encoding

How information is encoded depends on its original format and on the format used by the communication hardware.

As is known, information can be of two types, digital or analog, and signals can be of two types, also digital or analog. Therefore, four types of encoding are possible: digital-to-digital, analog-to-digital, digital-to-analog, and analog-to-analog.

- Digital-to-digital encoding is the representation of digital information by a digital signal.

- Analog-to-digital encoding is the representation of analog information by a digital signal.
- Digital-to-analog encoding is the representation of digital information by an analog signal.
- Analog-to-analog encoding is the representation of analog information by an analog signal. Radio is an example of an analog-to-analog communication.

11.1.3 Transmission Mode

The term “transmission mode” is used to define the direction of signal flow between two linked devices. There are three types of transmission modes: simplex, half-duplex, and full-duplex.

1. Simplex

In simplex mode, the communication is unidirectional, as one one-way street. Only one of the two stations on a link can transmit; the other can only receive.

Keyboards and traditional monitors are both examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output.

2. Half-Duplex

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa.

In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkie-talkies and CB radios are both half-duplex systems.

3. Full-Duplex

In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously.

The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in either direction share the capacity of the link. This sharing can occur in two ways: either the link must contain two physically separate transmission paths, one for sending the other for receiving, or the capacity of the channel is divided between signals traveling in opposite directions.

11.2 Introduction to Computer Networks

Due to rapid technological progress, these areas are rapidly converging, and the differences between collecting, transporting, storing, and processing information are quickly disappearing. As our ability to gather, process, and distribute information grows, the demand for even more sophisticated information processing grows even faster.

The merging of computers and communications has had a profound influence on the way computer systems are organized. The concept of the “computer center” as a room with a large computer to which users bring their work for processing is now totally obsolete. The old model of

a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called computer networks.

A network is a set of devices (often referred to as nodes) connected by media links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network. The links connecting the devices are often called communication channels.

We will use the term "computer networks" to mean an interconnected collection of autonomous computers. Two computers are said to being interconnected if they are able to exchange information. The connection need not be via a copper wire; fiber optics, microwaves, and communication satellites can also be used. By requiring the computers to be autonomous, we wish to exclude from our definition systems in which there is a clear master/slave relation. If one computer can forcibly start, stop, or control another one, the computers are not autonomous. A system with one control unit and many slaves is not a network; nor is a large computer with remote printers and terminals.

Connectivity is a concept related to using computer networks to link people and resources. For example, connectivity means that you can connect your microcomputer by telephone or other telecommunications links to other computers and information sources almost anywhere.

Clearly, different types of channels (cable or air) allow different kinds of networks to be formed. Telephone lines, for instance, may connect communications equipment within the same building or within a home. Networks may also be citywide and even international, using both cable and air connections.

11.3 Applications of Computer Networks

Computer networks have become an indispensable part of business, industry and entertainment in the short time they have been around. Some of the network applications in different fields are as follows:

1. Communicating

Communicating is by far the most popular Internet activity. You can exchange E-mail with your family and friends almost anywhere in the world. You can join and listen to discussions and debates on a wide variety of special-interest topics. You can even create your own personal Web page for friends and family to visit.

2. Business

Business or shopping is one of the fastest growing Internet applications. You can visit individual stores or a cybermall, which provides access to a variety of different stores. You can window shop, look for the latest fashions, search for bargains, and make purchases. You can purchase goods using checks, credit cards, or electronic cash.

3. Searching

Searching for information has never been more convenient. You can access some of the world's largest libraries directly from your home computer. You can visit virtual libraries, search through their stacks, read selected items, and even check out books. You will also find the latest local, national, and international news. Most newspapers maintain an online presence and presentations related to current news stories.

4. Entertainment

Entertainment options are nearly endless. You can find music, movies, magazines, and computer games. You will find live concerts, movie previews, book clubs, and interactive live games.

5. Education

Education or e-learning is another rapidly emerging Web application. You can take classes on almost any subject. There are courses just for fun and there are courses for high school, college, and graduate school credit. Some cost nothing to take and others cost a lot.

6. Manufacturing

Two applications that use networks to provide essential services are computer-aided design and computer-aided manufacturing, both of which allow multiple users to work on a project simultaneously.

11.4 Categories of Networks

Network architecture describes how a network is arranged and how resources are coordinated and shared. It encompasses a variety of different network specifics, including network configurations and strategies. Network configurations describe the physical arrangement of the network. Network strategies define how information and resources are shared.

11.4.1 Configurations

A network can be arranged or configured in several different ways. This arrangement is called the network's topology. The four principal network topologies are star, bus, ring, and hierarchical.

1. Star Network

In a star network, a number of small computers or peripheral devices are linked to a central unit. The central unit is the network hub and is typically a host computer or file server.

All communications pass through this central unit. Control is maintained by polling. That is, each connecting device is asked ("polled") whether it has a message to send. Each device is then in turn allowed to send its message.

One particular advantage of the star topology is that it can be used to support a time-sharing

system. That is, several users can share resources (time) on a central computer. The star is a common topology for linking microcomputers to a mainframe that allows access to an organization's database.

2. Bus Network

In a bus network each device in the network handles its own communications control. There is no host computer. All communications travel along a common connecting cable called a bus or backbone. As the information passes along the bus, it is examined by each device to see if the information is intended for it. The bus network is typically used when only a few microcomputers are to be linked together. This arrangement is common for sharing data stored on different microcomputers. Because a star network typically provides a more direct path to shared resources, it is more efficient than a bus network for sharing these resources. However, a bus network is easy to install and is less expensive.

3. Ring Network

In a ring network, each device is connected to two other devices, forming a ring. There is no central file server or computer. Messages are passed around the ring until they reach the correct destination. With microcomputers, the ring arrangement is the least frequently used of the four networks. However, it is often used to link mainframes, especially over wide geographical areas. These mainframes tend to operate fairly autonomously. They perform most or all of their own processing and only occasionally share data and programs with other mainframes.

A ring network is useful in a decentralized organization because it makes possible a distributed data processing system; that is, computers can perform processing tasks at their own dispersed locations. However, they also can share programs, data, and resources with each other.

4. Hierarchical Network

The hierarchical network, also called a hybrid network, consists of several computers linked to a central host computer, just like a star network. However, these computers are also hosts to others, smaller computers or to peripheral devices.

Thus, the host at the top of the hierarchy could be a mainframe. The computers below the mainframe could be minicomputers, and those below, microcomputers. The hierarchical network allows various computers to share databases, processing power, and different output devices.

A hierarchical network is useful in centralized organizations. For example, different departments within an organization may have individual microcomputers connected to departmental minicomputers. The minicomputers in turn may be connected to the corporation's mainframe, which contains data and programs accessible to all.

11.4.2 Strategies

Every network has a strategy, or way of coordinating the sharing of information and resources.

The most common network strategies are terminal, client/server, and peer-to-peer systems.

In a terminal network system, processing power is centralized in one large computer, usually a mainframe. The nodes connected to this host computer are either terminals with little or no processing capabilities or microcomputers running special software that allows them to act as terminals. The star and hierarchical networks are typical configurations with UNIX as the operating system.

Many airline reservation systems are terminal systems. A large central computer maintains all the airline schedules, rates, seat availability, and so on. Travel agents use terminals to connect to the central computer and to schedule reservations. Although the tickets may be primed along with travel itineraries at the agent's desk, nearly all processing is done at the central computer.

One advantage of terminal network systems is the centralized location and control of technical personnel, software, and data. One disadvantage is the lack of control and flexibility for the end user. Another disadvantage is that terminal systems do not use the full processing power available with microcomputers. Though the terminal strategy was once very popular, most new systems do not use it.

Client/server network systems use one computer to coordinate and supply services to other nodes on the network. The server provides access to resources such as Web pages, databases, application software, and hardware. This strategy is based on specialization. Server nodes coordinate and supply specialized services, and client nodes request the services. Commonly used network operating systems are Novell's NetWare, Microsoft's Windows NT, IBM's LAN Server, and Banyan Vines.

Client/server network systems are widely used on the Internet. For example, Napster (the once popular music service) employed a version of this strategy. Music enthusiasts used the Internet to connect to Napster servers. The Napster servers provided lists of music files (some of which were copyrighted) that were available to be copied from participating Napster users. The music enthusiasts were clients requesting services (information regarding the location of others willing to share music files) from Napster servers.

One advantage of the client/server network strategy is the ability to handle very large networks efficiently. Another advantage is the availability of powerful network management software to monitor and control network activities. The major disadvantages are the cost of installation and maintenance.

In a peer-to-peer network system, nodes have equal authority and can act as both clients and servers. For example, one microcomputer can obtain files located on another microcomputer and can also provide files to other microcomputers. A typical configuration for a peer-to-peer system is the bus network. Commonly used network operating systems are Novell's NetWare Lite, Microsoft's Windows NT, and Apple's Macintosh Peer-to-Peer LANs.

Many current popular music sharing services use this network strategy. In fact, the Napster approach was actually a hybrid network in which the Napster server worked in a client/server

environment providing a service to clients. Once a Napster user had the location of requested music files, he or she could sign off the network and then connect directly to the source forming a very simple peer-to-peer network. Each node could act as a server by providing access to music files and a client by receiving copies of music files. Today, Gnutella is a widely used peer-to-peer network system for sharing all kinds of files, including music files. Unlike the Napster approach, Gnutella networks directly connect users without a central server acting as the focal point for operations.

There are several advantages to the peer-to-peer network strategy. The networks are inexpensive and easy to install, and they usually work well for smaller systems with fewer than 10 nodes. Unlike the client/server network strategy, network operations are not dependent upon a single central node. As the number of nodes increases, however, the performance of the network declines. Another disadvantage is the lack of powerful management software to effectively monitor a large network's activities. For these reasons, peer-to-peer networks are typically used by smaller networks within organizations and for sharing files on the Internet.

11.4.3 LANs, MANs, and WANs

An alternative criterion for classifying networks is their scale. They can be divided into local, metropolitan, and wide area networks by their physical size. Finally, the connection of two or more networks is called an internetwork. The worldwide Internet is a well-known example of an internetwork.

1. Local Area Networks

Local area networks, generally called LANs, are privately owned networks within a single building or campus of up to a few kilometers in size. They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information. LANs are distinguished from other kinds of networks by three characteristics: their size, their transmission technology, and their topology.

LANs often use a transmission technology consisting of a single cable to which all the machines are attached. Traditional LANs run at speeds of 10 to 100 Mbps, have low delay (tens of microseconds), and make very few errors. Today, however, speeds are increasing and can reach hundreds of megabits/sec with gigabit systems in development.

In general, a given LAN will use only one type of transmission medium. Various topologies are possible for LANs. The most common LAN topologies are bus, ring, and star.

2. Metropolitan Area Networks

A metropolitan area network, or MAN (plural: MANs, not MEN) is basically a bigger version of a LAN and normally uses similar technology.

MAN is designed to extend over an entire city. It may be a single network such as a cable television network, or it may be a means of connecting a number of LANs into a larger network so

that resources may be shared LAN-to-LAN as well as device-to-device. For example, a company can use a MAN to connect the LANs in all of its offices throughout a city.

3. Wide Area Networks

A wide area network, or WAN, spans a large geographical area that may comprise a country, a continent, or even the whole world. It provides long-distance transmission of data, voice, image, and video information over large geographical areas.

In contrast to LANs (which depend on their own hardware for transmission), WANs may utilize public, leased, or private communication devices, usually in combinations, and can therefore span an unlimited number of miles.

A WAN that is wholly owned and used by a single company is often referred to as an enterprise network.

4. Internetworks

Many networks exist in the world, often with different hardware and software. People connected to one network often want to communicate with people attached to a different one. This desire requires connecting together different, and frequently incompatible networks, sometimes by using machines called gateways to make the connection and provide the necessary translation, both in terms of hardware and software. A collection of interconnected networks is called an internetworks or just internet.

When two or more networks are connected, they become an internetwork. Individual networks are joined into internetworks by the use of internetworking devices. These devices include routers and gateways. The term "internet" (lowercase i) should not be confused with the Internet (uppercase I). The first is a generic term used to mean an interconnection of networks. The second is the name of a specific worldwide network.

11.4.4 Intranets and Extranets

Computer networks in organizations have evolved over time. Most large organizations have a complex and wide range of different network configurations, operating systems, and strategies. Integrating or connecting all of these networks has been a very challenging task. One way is to apply Internet technologies to support communication within and between organizations using intranets and extranets.

1. Intranets

An intranet is a private network within an organization that resembles the Internet. Like the public Internet, intranets use browsers, Web sites, and Web pages. Intranets typically provide E-mail, mailing lists, newsgroups, and FTP services accessible only to those within the organization.

Organizations use intranets to provide information to their employees. Typical applications include electronic telephone directories, E-mail addresses, employee benefit information, internal

job openings, and much more. Employees find surfing their organizational intranets to be as easy and as intuitive as surfing the Internet.

2. Extranets

An extranet is a private network that connects more than one organization. Many organizations use Internet technologies to allow suppliers and others limited access to their networks. The purpose is to increase efficiency and reduce costs. For example, General Motors has thousands of suppliers for the parts that go into making an automobile. By having access to the production schedules, suppliers can schedule and deliver parts as they are needed at the General Motors assembly plants. In this way, General Motors can be assured of having adequate parts without maintaining large inventories.

11.5 OSI and TCP/IP Reference Model

11.5.1 OSI Reference Model

The standard model for networking protocols and distributed applications is the International Standard Organization's Open System Interconnect (ISO/OSI) model (Figure 11-1). It defines seven network layers.

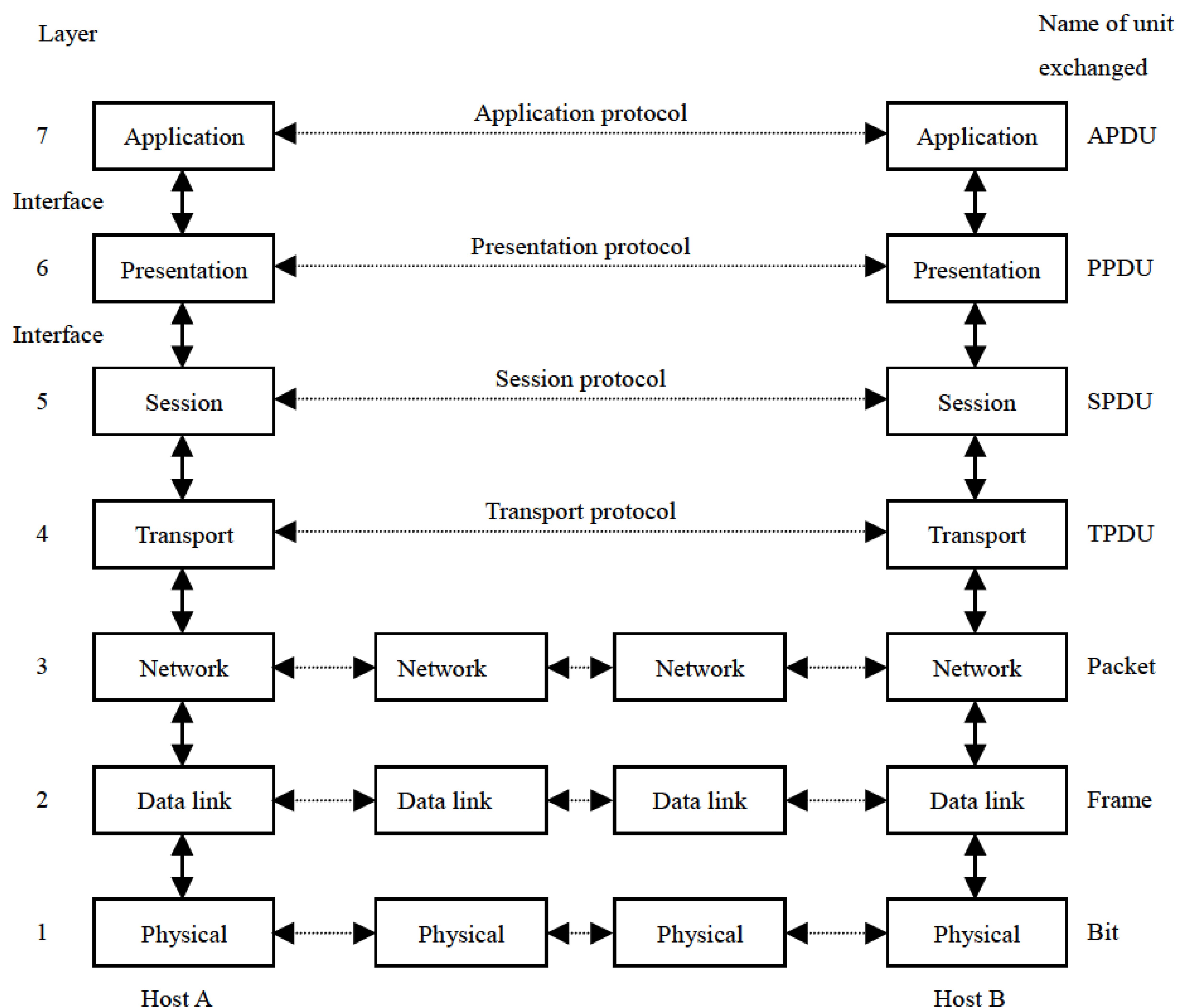


Figure 11-1 the OSI reference model

The principles that were applied to arrive at the seven layers are as follows:

- A layer should be created where a different level of abstraction is needed.
- Each layer should perform a well defined function.
- The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
- The layer boundaries should be chosen to minimize the information flow across the interfaces.
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy.

1. Physical Layer

Physical layer defines the cable or physical medium itself, e.g., thinnet, thicknet, unshielded twisted pairs (UTP). All media are functionally equivalent. The main difference is in convenience and cost of installation and maintenance. Converters from one media to another operate at this level.

2. Data Link Layer

Data Link layer defines the format of data on the network. A network data frame, aka packet, includes checksum, source and destination address, and data. The largest packet that can be sent through a data link layer defines the Maximum Transmission Unit (MTU). The data link layer handles the physical and logical connections to the packet's destination, using a network interface. A host connected to an Ethernet would have an Ethernet interface to handle connections to the outside world, and a loopback interface to send packets to itself.

Ethernet addresses a host using a unique, 48-bit address called its Ethernet address or Media Access Control (MAC) address. MAC addresses are usually represented as six colon-separated pairs of hex digits, e.g., 8:0:20:11:ac:85. This number is unique and is associated with a particular Ethernet device. Hosts with multiple network interfaces should use the same MAC address on each. The data link layer's protocol-specific header specifies the MAC address of the packet's source and destination. When a packet is sent to all hosts (broadcast), a special MAC address (ff:ff:ff:ff:ff:ff) is used.

3. Network Layer

NFS uses Internetwork Protocol (IP) as its network layer interface. IP is responsible for routing, directing datagrams from one network to another. The network layer may have to break large datagrams, larger than MTU, into smaller packets and host receiving the packet will have to reassemble the fragmented datagram. The Internetwork Protocol identifies each host with a 32-bit IP address. IP addresses are written as four dot-separated decimal numbers between 0 and 255, e.g., 129.79.16.40. The leading 1-3 bytes of the IP identify the network and the remaining bytes identify the host on that network. The network portion of the IP is assigned by InterNIC Registration

Services, under the contract to the National Science Foundation, and the host portion of the IP is assigned by the local network administrators.

Even though IP packets are addressed using IP addresses, hardware addresses must be used to actually transport data from one host to another. The Address Resolution Protocol (ARP) is used to map the IP address to it hardware address.

4. Transport Layer

Transport layer subdivides user-buffer into network-buffer sized datagrams and enforces desired transmission control. Two transport protocols, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), sit at the transport layer. Reliability and speed are the primary difference between these two protocols. TCP establishes connections between two hosts on the network through “sockets” which are determined by the IP address and port number. TCP keeps track of the packet delivery order and the packets that must be resent. Maintaining this information for each connection makes TCP a stateful protocol. UDP on the other hand provides a low overhead transmission service, but with less error checking. NFS is built on top of UDP because of its speed and statelessness. Statelessness simplifies the crash recovery.

5. Session Layer

The session protocol defines the format of the data sent over the connections. The NFS uses the Remote Procedure Call (RPC) for its session protocol. RPC may be built on either TCP or UDP. Login sessions use TCP whereas NFS and broadcast use UDP.

6. Presentation Layer

External Data Representation (XDR) sits at the presentation level. It converts local representation of data to its canonical form and vice versa. The canonical uses a standard byte ordering and structure packing convention, independent of the host.

7. Application Layer

Application layer provides network services to the end-users. Mail, FTP, Telnet, DNS, NIS, NFS are examples of network applications.

11.5.2 TCP/IP Reference Model

Although the OSI model is widely used and often cited as the standard, TCP/IP protocol has been used by most Unix workstation vendors. TCP/IP is designed around a simple four-layer scheme. It does omit some features found under the OSI model. Also it combines the features of some adjacent OSI layers and splits other layers apart. The four network layers defined by TCP/IP model are as follows.

- Host-to-Network Layer. This layer defines the network hardware and device drivers.
- Internet Layer. This layer is used for basic communication, addressing and routing. TCP/IP uses IP and ICMP protocols at the network layer.

- Transport Layer. This layer handles communication among programs on a network. TCP and UDP fall into this layer.
- Application Layer. End-user applications reside at this layer. Commonly used applications include NFS, DNS, ARP, talk, FTP, ntp, and traceroute.

Technical Notes to the Text

1. bandwidth，带宽。带宽指信道上能够传输信号的最大频率范围。

2. bit rate，比特率，位速率。其指二进制数据在一条通信线路上的传输速度，通常用每秒多少位(bps)来表示。

3. baud rate，波特率。其指每秒钟能够传送的信息位的数量，是所传送的代码的最短码元占有时间的倒数。

4. simplex，单工。单工指信号只能按一个方向传播，任何时候都不能改变信号的传播方向，如电视信号。

5. half-duplex，半双工。半双工指信号可以双向传送，但必须交替进行，一个时间只能向一个方向传送，如电话线路。

6. full-duplex，全双工。全双工指信号可以同时双向传送，如计算机之间的通信。

7. computer network，计算机网络。计算机网络就是把分布在不同地点的、具有独立功能的多个计算机物理地连接起来，按照网络协议相互通信，以共享软件、硬件和数据资源为目标的系统。

8. star network，星型网络。这种网络中的所有计算机都连接到一个中心设备，计算机间的通信都需要通过中心设备进行转发。

9. bus network，总线型网络。这种网络中的所有设备都连接到一条公共的总线，所有通信都通过总线进行传输。

10. ring network，环型网络。在这种网络中，每个设备都与相邻设备相连，形成一个环状结构，报文沿着环进行传输。

11. hierarchical network，分层网络。这种网络中的下层计算机以星型结构连接到上层的主机，形成以一台中心主机为根的树形分层结构。

12. client/server network system，客户机/服务器网络系统。这种网络系统中使用一台计算机(称为服务器)来为其他计算机(称为客户机)提供特定的服务。

13. peer-to-peer network system，对等网络系统。这种网络中的计算机地位平等，既可以作为服务器提供服务，也可以作为客户机来获得其他计算机的服务。

14. Local Area Network (LAN)，局域网。局域网是一种连接办公室电子设备，并在一间办公室或一座建筑物内构成一个网络的系统。

15. Metropolitan Area Network (MAN)，城域网。在 5~100km 的地理覆盖范围内，以高传输速率充分支持数据、声音和图像等综合业务传输的一种通信网络。

16. Wide Area Network (WAN)，广域网。广域网也称远程网，是一种作用的地理范围从数十千米到数千千米，可以连接若干个城市、地区，甚至跨国界的遍及全球的通信网络。

17. Internet，因特网。因特网指将分布在不同地理位置的网络、设备相连接，以构成更大规模的互联网络系统，实现互联网络的共享。

18. Intranet，企业内部网。企业内部网是一个组织或机构内的私有网络，这种网络也采用与互联网相同的技术和服务为员工提供信息。

19. Extranet，企业外部网。企业外部网是连接多个组织或机构的私有网络，这种网络也采用互联网技术为合作机构提供有限的服务。

20. OSI (open systems interconnection) reference model，开放系统互联参考模型。该模型定义了异种机联网标准的框架结构。其中，“系统”指计算机、外部设备、终端、传输设备、操作人员及相应软件；“开放”指按照参考模型建立的任意两个系统之间的连接或操作。当一个系统能按 OSI 模型与另一个系统进行通信时，就称该系统为开放系统。

21. physical layer，物理层。该层实现网络设备之间的物理连接，其主要功能是传输 0 或 1 的二进制位的数据流。物理层规定了各种设备的电气特征。

22. data link layer，数据链路层。该层以帧为单位提供一种可靠的传输数据的方法。

23. network layer，网络层。该层实现路由选择和控制信息的中间转发，如果目标设备在另一个网络中，那么该层将决定数据通过何种路由到达目的地，并将报文形式的数据转换成数据包，经过一些中间设备从一个网络传送到另一个网络。

24. transport layer，传输层。该层为一对通信用户建立端点到端点的连接，并提供数据传输服务。该层还有监控功能，确保信息成功到达目标设备。

25. session layer，会话层。该层组织和管理两个用户之间的对话，规定网络物理地址与逻辑地址的转换。

26. presentation layer，表示层。该层用于不同信息表示方法之间的转换、协商和建立数据交换的格式，以及数据压缩、报文的加密和解密。

27. application layer，应用层。该层是用户的应用程序与网络之间的接口，为用户提供一个开放系统互联的工作环境。

28. TCP/IP (transmission control protocol/internet protocol) reference model，TCP/IP 参考模型。TCP/IP 协议是在网络中提供可靠数据传输和无连接数据报服务的一组协议。提供可靠数据传输的协议称为传输控制协议(TCP)，提供无连接数据报服务的协议称为网际协议(IP)。TCP/IP 协议已广泛用于因特网。

29. UDP (user datagram protocol)，用户数据报协议。它是一个不可靠的、无连接的协议，用于不需要 TCP 的排序和流量控制能力而是自行完成这些功能的应用程序。它被广泛地应用于只有一次的、客户/服务器模式的请求—应答查询，以及快速递交比准确递交更重要的应用程序，如传输语音或影像。

Word Bank to the Text

A. Useful new words

dominate *v.* 统治，支配

accompany	*v.* 陪伴，伴奏
electromagnetic	*adj.* 电磁的
discrete	*adj.* 不连续的，离散的
spectrum	*n.* 光谱，频谱
predetermine	*v.* 预定，预先确定
interrogation	*n.* 审问，问号
reservation	*n.* 保留，预定
unidirectional	*adj.* 单向的，单向性的
simultaneously	*adv.* 同时地
sophisticated	*adj.* 高度发展的，精密复杂的
equivalent	*adj.* 相等的，相当的
incompatible	*adj.* 性质相反的，矛盾的
merge	*v.* 合并，并入
profound	*adj.* 深刻的，意义深远的，渊博的，造诣深的
obsolete	*adj.* 荒废的，陈旧的
autonomous	*adj.* 自治的，自我管理的
indispensable	*adj.* 不可缺少的，绝对必要的
cybermall	*n.* 网络商城
encompass	*v.* 包括，包含
topology	*n.* 拓扑，布局
peripheral	*adj.* 外围的　*n.* 外围设备
decentralize	*v.* 疏散，分散
disperse	*v.* (使)分散，(使)散开
hybrid	*adj.* 混合的
hierarchical	*adj.* 分等级的
itinerary	*n.* 路线
flexibility	*n.* 弹性，适应性
participate	*v.* 参与，参加
enthusiast	*n.* 狂热者
maintenance	*n.* 维护，保持
metropolitan	*adj.* 大城市的，大都会的
unwieldy	*adj.* 笨拙的，不实用的
resemble	*v.* 像，类似
intuitive	*adj.* 直觉的
unshield	*adj.* 无保护的
fragment	*v.* 使成为碎片
enforce	*v.* 强迫，执行
canonical	*adj.* 规范的

omit	*v.* 省略，疏忽
adjacent	*adj.* 邻近的，接近的
reside	*v.* 居住，驻留

B. Useful expressions

a stream of	一连串的
turn into	进入，(使)变成
separate from	分离，分开
at intervals	不时，相隔一定距离
take actions to	对……采取行动
vice versa	反之亦然
have a profound influence on	对……具有深刻的影响
by far	到目前为止
a variety of	多种的
convert into	转变，变换
search for	搜查，搜索
access from	存取，接近
for fun	当作玩笑，不是认真的
be intended for	打算供……使用
reach the destination	达到目的
along with	连同……一起，随同……一起
act as	担当，扮演……角色
distinguish from	区别，识别
in contrast to	和……形成对比
join into	加入
be confused with	混淆
surf the Internet	浏览网络
be assured of	确认，有信心
perform a function	履行职责
with an eye toward	着眼于
split apart	分裂开
cite as	引用，引证

C. Technical terms and proper names

analog and digital	模拟和数字
frequency	频率
spectrum	频谱
bandwidth	带宽
bit rate	比特率

baud rate	波特率
simplex	单工
half-duplex	半双工
full-duplex	全双工
computer network	计算机网络
star network	星形网络
bus network	总线型网络
ring network	环形网络
hierarchical network	分层网络
client/server network system	客户机/服务器网络系统
peer-to-peer network system	对等网络系统
Local Area Network (LAN)	局域网
Metropolitan Area Network (MAN)	城域网
Wide Area Network (WAN)	广域网
Internet	因特网
OSI reference model	开放系统互联参考模型
physical layer	物理层
data link layer	数据链路层
network layer	网络层
transport layer	传输层
session layer	会话层
presentation layer	表示层
application layer	应用层
TCP/IP reference model	TCP/IP 参考模型
host-to-network layer	主机-网络层
Internet layer	网际层
IP	互联网协议
TCP	传输控制协议
UDP	用户数据报协议

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. A ________ is a set of devices (often referred to as nodes) connected by media links. A node can be a computer, printer, or any other device capable of sending and/or receiving data. The links connecting the devices are often called ________________.

2. In the transport layer of the TCP/IP model, two end-to-end protocols have been defined. They are ________ and ________.

3. There are three types of network strategy: terminal, ________ and ________.

4. Networks can be divided into ________, ________, and ________ by their scale.

5. ________ is the number of bits transmitted during one second. ______ refers to the number of signal units per second that are required to represent those bits.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. Signals can be analog or digital. Analog signals have only a limited number of values, but digital signals can have any value in a range. ( )

2. The OSI model has seven layers and the TCP/IP has four layers. Both have (inter)network, data link, and application layers, but the other layers are different. ( )

3. The OSI model supports both connectionless and connection-oriented communication in the transport layer. ( )

4. The advantage of the peer-to-peer network strategy is the ability to handle large networks and the availability of powerful network management software. ( )

5. Reliability and speed are the primary difference between TCP and UDP. ( )

Ⅲ. Match each of the following terms to the appropriate definition.

simplex	application layer	full-duplex

1. ________ In this mode, the communication is unidirectional, as one one-way street. Only one of the two stations on a link can transmit; the other can only receive.

2. ________ In this mode, both stations can transmit and receive simultaneously.

3. ________ To allow access to network resources.

Ⅳ. List four primary physical network configurations.

1. ________________ 2. ________________

3. ________________ 4. ________________

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

bandwidth	accompany	resemble
equivalent	incompatible	participate
adjacent	dominate	predetermine

1. The book is expected to ________ the best-seller lists.

2. The proposal was instantly voted through with two to one in favor, ________ by enthusiastic applause.

3. Many institutions exchange information by hand because of ________ computer systems.

4. The schools are ________ but there are separate doors.

5. Even the cheapest car costs the ________ of 70 years' salary for a government worker.

6. It is widely accepted that a person's health is often genetically ________.

7. The young pretty English teacher ________ her brother in looks.

8. Union leaders called for the active ________ of all members in the day of protest.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "attach" and fill in the sentences with the right word.

Example: LANs often use a transmission technology consisting of a single cable to which all the **machines** are **attached**.

1. In order to improve people's living standard, we should attach great im______ to the development of the economy.

2. Different people attach different me______ to words.

3. The workers at the airport attach la______ to the luggage of the passengers.

4. The experts wouldn't attach too much we______ to these findings.

5. After the fourth round of negotiations the both sides agreed to attach a st______ to the contract.

6. Some clever hackers can use tools to attach the vi______ to legitimate programs such as cartoons or greetings.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

A computer networks is an interconnected group of computers. Networks may be __1__ by the network layer at which they operate according to basic reference models considered __2__ standards in the industry. __3__ the seven-layer Open Systems Interconnection (OSI) reference model is better known in academia, the __4__ of networks use the Internet Protocol Suite (IP).

Computer networks may be classified according to the __5__: Personal Area Network (PAN), __6__ Area Network (LAN), Campus Area Network (CAN), Metropolitan Area Network (MAN), or Wide Area Network (WAN). Computer networks can also be classified according to the hardware technology that is used to connect the __7__ devices in the network such as optical fibre, Ethernet, Wireless LAN, HomePNA, or Power line communication.

Network Topology signifies the way in which intelligent devices in the network see their logical relations to __8__. Network topology is __9__ of the "physical" layout of the network. In this regard the visual and operational characteristics of a network are distinct; the logical network topology is not __10__ the same as the physical layout.

1. A. grouped	B. classified	C. defined	D. graded
2. A. as	B. be	C. of	D. to
3. A. When	B. Since	C. While	D. Now that
4. A. majority	B. major	C. more	D. maximum
5. A. size	B. scale	C. expense	D. cost
6. A. Local	B. Location	C. Listed	D. Lined
7. A. special	B. most	C. individual	D. peripheral
8. A. each other	B. every one	C. the other	D. one another
9. A. dependent	B. depending	C. independent	D. independently
10. A. necessarily	B. unnecessarily	C. necessary	D. unnecessary

Translation

Ⅷ. Translate the following into Chinese.

1. Organizations with hundreds of offices spread over a wide geographical area routinely expect to be able to examine the current status of even their most remote outpost at the push of a button.

2. The old model of a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job.

3. Technological advances are making it possible for communications links to carry more and faster signals. As a result, services are evolving to allow use of the expanded capacity, including extensions to established telephone service.

4. There is no generally accepted taxonomy into which all computer networks fit, but two dimensions stand out as important: transmission technology and scale.

5. ISO's purpose is to promote the development of standardization and related activities to facilitate international exchange of goods and services.

Chapter 12
LAN, MAN & WAN

Pre-reading Questions

1. What is IEEE 802?
2. What is CSMA/CD?
3. Compare and contrast Ethernet with Token Ring.

12.1 Local Area Networks (LANs)

Networks with nodes that are in close physical proximity — within the same building, for instance — are called local area networks. Typically, LANs span distances less than a mile and are owned and operated by individual organizations. LANs are widely used by colleges, universities, and other types of organizations to link microcomputers and to share printers and other resources.

The LAN is a typical arrangement and provides two benefits: economy and flexibility. People can share costly equipment. For instance, the four microcomputers share the laser printer and the file server, which are expensive pieces of hardware. Other equipment or nodes may also be added to the LAN — for instance, more microcomputers, a mainframe computer, or optical-disk storage devices. Additionally, the network gateway is a device that allows one LAN to be linked to other LANs or to larger networks. For example, the LAN of one office group may be connected to the LAN of another office group.

There are a variety of different standards or ways in which nodes can be connected to one another and ways in which their communications are controlled in a LAN. The most common standard is known as Ethernet. LANs using this standard are sometimes referred to as Ethernet LANs.

12.1.1 Ethernet (802.3)

Ethernet is a local area network (LAN) protocol. Recall that a protocol is a set of rules followed by all of the computers in a given network so that data being transferred do not collide into each other. Ethernet was developed in the 1970s by Xerox Corporation. Then in about 1979, three companies, Xerox, Digital Equipment Corporation, and Intel Corporation, got together to try and standardize this new system. The result of this attempt is called DIX (Digital, Intel, Xerox). It is what is commonly referred to as "Ethernet". However, in 1983, a new standard for Ethernet was created — the Institute of Electrical and Electronic Engineers (IEEE) released the Ethernet standard called IEEE 802.3.

Ever since its introduction to the world, Ethernet has become increasingly popular. It is now one of the most common LAN protocols used throughout the world!

Recently, newer versions of Ethernet have been developed. They are much faster than the original Ethernets. One of the newest versions, Fast Ethernet, can transfer data up to 100 megabits (100 million bits) per second! There is also the Gigabit Ethernet, which transfers 1 gigabit (1 billion bits) of data per second.

802.3 defines two categories: baseband and broadband, as shown in Figure 12-1. The word base specifies a digital signal (in this case, Manchester encoding). The word broad specifies an analog signal (in this case, PSK encoding). IEEE divides the baseband category into five different standards (see Table 12-1): 10BASE5 (Thick Ethernet), 10BASE2 (Thin Ethernet), 10BASE-T (Twisted-pair Ethernet), 1BASE5 (StarLAN), and 100BASE-T (Fast Ethernet). The first number (10, 1, or 100) indicates data rate in Mbps. The last number or letter (5, 2, 1, or T) indicates maximum cable length or the type of cable. The maximum cable length restriction can be changed using networking devices such as repeaters or bridges. For example, the notation 10BASE5 means that it operates at 10 Mbps, uses baseband signaling, and can support segments of up to 500 meters. IEEE defines only one specification for the broadband category: 10BROAD36 (broadband Ethernet).

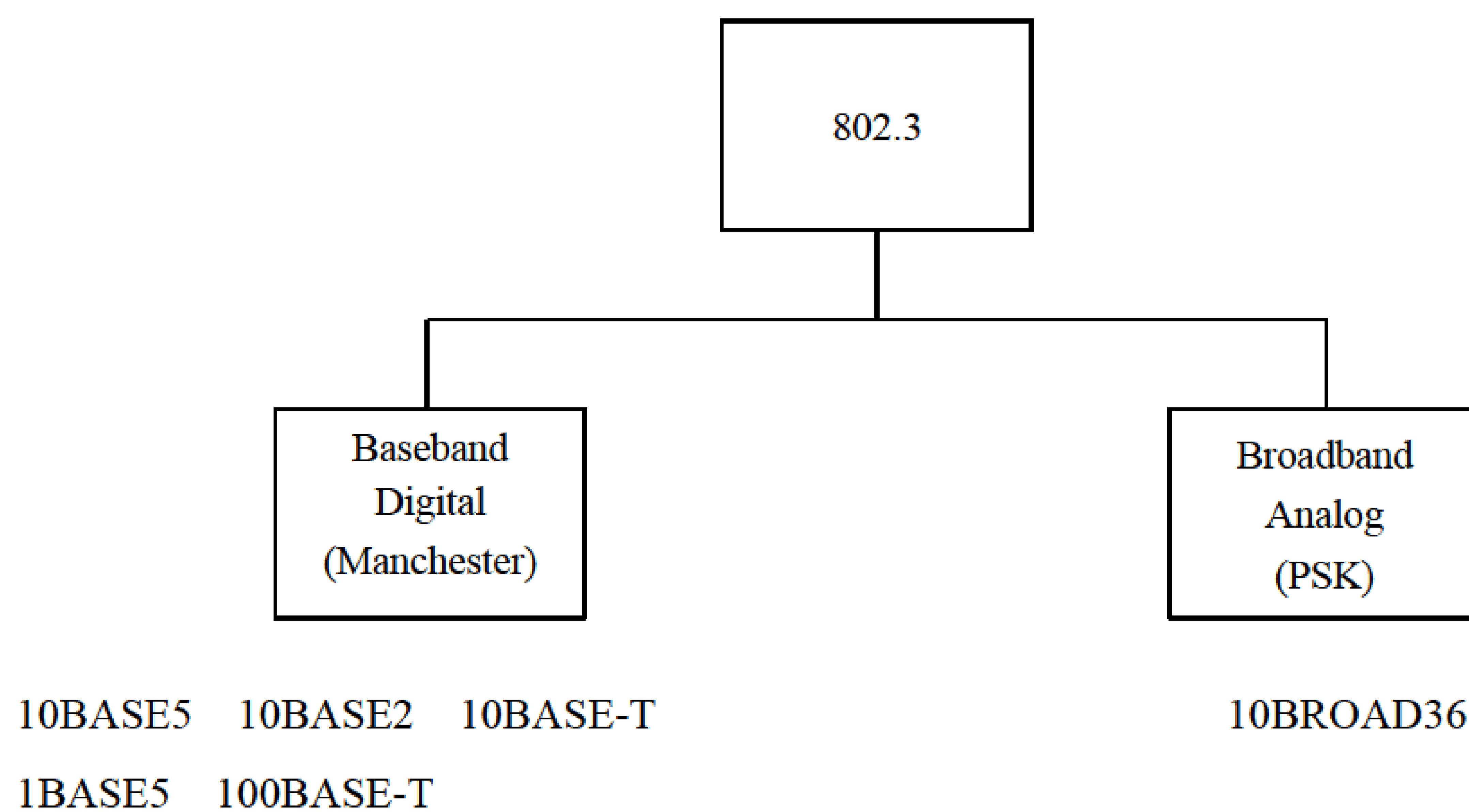


Figure12-1 Two Categories of 802.3

Table 12-1 The Most Common Kinds of Baseband 802.3 LANs

Name	Cable	Data Rate (Mbps)	Max. segment (m)
10BASE5	Thick coax	10	500
10BASE2	Thin coax	10	185
10BASE-T	Twisted pair	10	100
1BASE5	Twisted pair	1	500
100BASE-T	Twisted pair	100	100

Whenever multiple users have unregulated access to a single line, there is a danger of signals overlapping and destroying each other. Such overlaps, which turn the signals into unusable noise,

are called collisions. As traffic increases on a multiple access link, so do collisions. A LAN therefore needs a mechanism to coordinate traffic, minimize the number of collisions that occur, and maximize the number of frames that are delivered successfully. The access mechanism used in an Ethernet is called carrier sense multiple access with collision detection (CSMA/CD, standardized in IEEE802.3).

CSMA/CD is the result of an evolution from multiple access (MA) to carrier sense multiple access (CSMA), and finally, to carrier sense multiple access with collision detection (CSMA/CD). The original design was a multiple access method in which every workstation had equal access to a link.

The final step is the addition of collision detection (CD). In the CSMA/CD the station wishing to transmit first listens to make certain the link is free, then transmits its data, then listens again. During the data transmission, the station checks the line for the extremely high voltages that indicate a collision. If a collision is detected, the station waits a predetermined amount of time for the line to clear, then sends its data again.

Today the terms Ethernet, CSMA/CD, and 802.3 are used interchangeably.

12.1.2 Token Bus (802.4)

Although 802.3 is widely used in offices, during the development of the 802 standard, people from General Motors and other companies interested in factory automation had serious reservations about it. For one thing, due to the probabilistic MAC protocol, with a little bad luck a station might have to wait arbitrarily long to send a frame (i.e., the worst case is unbounded). For another, 802.3 frames do not have priorities, making them unsuited for real-time systems in which important frames should not be held up waiting for unimportant frames.

A simple system with a known worst case is a ring because a break in the ring cable would bring the whole network down. Furthermore, they noted that a ring was a poor fit to the linear topology of most assembly lines. As a result, a new standard was developed, having the robustness of the 802.3 broadcast cables, but the known worst-case behavior of a ring.

This standard, 802.4, describes a LAN called a token bus. Physically, the token bus is a linear or tree-shaped cable onto which the stations are attached. Logically, the stations are organized into a ring, with each station knowing the address of the station to its “left” and “right”. When the logical ring is initialized, the highest numbered station may send the first frame. After it is done, it passes permission to its immediate neighbor by sending the neighbor a special control frame called a token. The token propagates around the logical ring, with only the token holder permitted to transmit frames. Since only one station at a time holds the token, collisions do not occur.

12.1.3 Token Ring (802.5)

Like Ethernet, a Token Ring is a local-area network (LAN) protocol. Developed by IBM, Token Ring was first introduced to the public in 1984. However, Token Ring is not as popular or as widely used as Ethernet.

Token ring requires that stations take turns sending data. Each station may transmit only during its turn and may send only one frame during each turn. The mechanism that coordinates this rotation is called token passing. A token is a simple placeholder frame that is passed from station to station around the ring.

Whenever the network is unoccupied, it circulates a simple three-byte token. This token is passed from NIC to NIC in sequence until it encounters a station with data to send. That station waits for the token to enter its network board. If the token is free, the station may then send a data frame. It keeps the token and sets a bit inside its NIC as a reminder that it has done so, then sends its one data frame.

This data frame proceeds around the ring, being regenerated by each station. Each intermediate station examines the destination address, finds that the frame is addressed to another station, and relays it to its neighbor. The intended recipient recognizes its own address, copies the message, checks for errors, and changes four bits in the last byte of the frame to indicate address recognized and frame copied. The full packet then continues around the ring until it returns to the station that sent it.

The sender receives the frame and recognizes itself in the source address field. It then examines the address-recognized bits. If they are set, it knows the frame was received. The sender then discards the used data frame and releases the token back to the ring.

1. Priority and Reservation

Each station has a priority code. As a frame passes by, a station waiting to transmit may reserve the next open token by entering its priority code in the access control (AC) field of the token or data frame (discussed later in this section). A station with a higher priority may remove a lower priority reservation and replace it with its own. Among stations of equal priority, the process is first come, first served. Through this mechanism, the station holding the reservation gets the opportunity to transmit as soon as the token is free, whether or not it comes next physically on the ring.

2. Time Limit

To keep traffic moving, token ring imposes a time limit on any station wanting to use the ring. A starting delimiter (the first field of either a token or data frame) must reach each station within a specified interval (usually 10 milliseconds).

3. Monitor Station

Several problems may occur to disrupt the operation of a token ring network. In one scenario, a station may neglect to retransmit a token or a token may be destroyed by noise, in which case there is no token on the ring and no station may send data. In another scenario, a sending station may neglect to remove its used data frame from the ring or may not release the token once its turn has ended.

To handle these occurrences, one station on the ring is designated as a monitor. The monitor sets a timer each time the token passes. If the token does not reappear in the allotted time, it is

presumed to be lost and the monitor generates a new token and introduces it to the ring.

12.1.4 Fiber Distributed Data Interface (FDDI)

FDDI is a local area network protocol standardized by ANSI and the ITU-T (ITU-T X.3). It supports data rates of 100 Mbps and provides a high-speed alternative to Ethernet and token ring. When FDDI was designed, speeds of 100 Mbps required fiber-optic cable. Today, however, comparable speeds are available using copper cable. The copper version of FDDI is known as CDDI.

1. Access Method: Token Passing

In a token ring network, a station can send only one frame each time it captures the token. In FDDI, access is limited by time. A station may send as many frames as it can within its allotted access period, with the proviso that time-sensitive frames are sent first.

To implement this access mechanism, FDDI differentiates between two types of data frames: synchronous and asynchronous. Synchronous here refers to information that is time sensitive, while asynchronous refers to information that is not. These frames are usually called S-frames and A-frames.

Each station that captures the token is required to send S-frames first. In fact, it must send its S-frames whether or not its time allotment has run out (see below). Any remaining time may then be used to send A-frames. To understand how this mechanism ensures fair and timely link access, it is necessary to understand the FDDI time register and timer.

2. Time Register

FDDI defines three time registers to control circulation of the token and distribute link access opportunities among the nodes equitably. Each station has three registers. The registers hold time values that control the operation of the ring. These values are set when the ring is initialized and do not vary in the course of operation. The registers are called synchronous allocation (SA), target token rotation time (TTRT), and absolute maximum time (AMT).

Synchronous Allocation (SA). The SA register indicates the length of time allowing each station for sending synchronous data. This value is different for each station and is negotiated during initialization of the ring.

Target Token Rotation Time (TTRT). The TTRT register indicates the average time required for a token to circulate around the ring exactly once (the elapsed time between a token’s arrival at a given station and its next arrival at the same station). This value is the same for all stations and is negotiated during the initialization of the ring. Because it is an average, the actual time of any rotation may be greater or less than this value.

Absolute Maximum Time (AMT). The AMT register holds a value equal to twice the TTRT. A token may not take longer than this time to make one rotation of the ring. If it does, some station or stations are monopolizing the network and the ring must be reinitialized.

3. Timer

Each station contains a set of timers that enable it to compare actual timings with the values contained in the registers. Timers can be set and reset, and their values decremented at a rate set by the system clock. The two timers used by FDDI are called the token rotation timer (TRT), and token holding timer (THT).

Token Rotation Timer (TRT). The TRT runs continuously and measures the actual time taken by the token to complete a cycle. When the token returns, the station records the time remaining on its TRT into its THT. Then the station resets its TRT based on the TTRT value. As soon as the TRT is set, it begins counting down. The time indicated by the TRT at any given point therefore is the difference between the actual time that has elapsed during the current rotation and the expected or allowed time (TTRT time). When the token completes a rotation and returns to the station, the time indicated by the TRT is equal to the amount of time remaining for that rotation (the difference between the TTRT and the actual elapsed time). That remaining time is then available to the station to send its frames.

Token Holding Timer (THT). The THT begins running as soon as the token is received. Its function is to show how much time remains for sending asynchronous frames once the synchronous frames have been sent. Each time the station receives the token, the TRT value is copied into the THT. At that point, the THT starts its own countdown. Any waiting synchronous frames must be sent as soon as the token is received. The THT shows how much time (if any) remains for sending asynchronous frames. The station may send only as many A-frames as it has THT credit for. As long as the THT is positive, the station can send asynchronous data. Once the value of this timer reaches or falls below zero, however, the station must release the token. We may think of the THT as the station’s bank account. S-frames are bills that must be paid immediately — even if the station has to go into debt to do so. A-frames are expenditures that can be put off for a while; the station needs to make them but can wait until its bank account can cover the expense.

12.1.5 Comparison

Table 12-2 compares the features of the three LANs discussed above. Ethernet is good for low-level loads but collapses as the load increases due to collisions and retransmissions. Token ring and FDDI perform poorly at low-level loads but always guarantee some maximum time between transmission of two adjacent frames.

Table 12-2 LAN Comparison

Network	Access Method	Signaling	Data rate	Error Control
Ethernet	CSMA/CD	Manchester	1~10Mbps	No
Token Ring	Token passing	Differential Manchester	10~16Mbps	Yes
FDDI	Token passing	4B/5B	100Mbps	Yes

12.2 Metropolitan Area Networks (IEEE 802.6)

For networks covering an entire city, IEEE defined one MAN, called DQDB (Distributed Queue Dural Bus), as standard 802.6. In this section we will examine how it works.

The basic geometry of 802.6 is two parallel, unidirectional buses. The two parallel, unidirectional buses snake through the city, with stations attached to both buses in parallel. Each bus has a head-end, which generates a steady stream of 53-byte cells. Each cell travels downstream from the head-end. When it reaches the end, it falls off the bus.

Each cell carries a 44-byte payload field, making it compatible with some AAL modes. Each cell also holds two protocol bits, Busy, set to indicate that a cell is occupied, and Request, which can be set when a station wants to make a request.

To transmit a cell, a station has to know whether the destination is to the left of it or to the right of it. If the destination is to the right, the sender uses bus A. Otherwise, it uses bus B. Unlike all the other 802 LAN protocols, in 802.6, stations queue up in the order they became ready to send and transmit in FIFO order.

12.3 Wide Area Networks

12.3.1 Introduction

A wide area network, or WAN, spans a large geographical area, often a country or continent. It contains a collection of machines intended for running user (i.e., application) programs.

In most WANs, the network contains numerous cables or telephone lines, each one connecting a pair of routers. If two routers that do not share a cable, nevertheless, wish to communicate, they must do this indirectly, via other routers. When a packet is sent from one router to another via one or more intermediate routers, the packet is received at each intermediate router in its entirety, stored there until the required output line is free, and then forwarded. A subnet using this principle is called a point-to-point, store-and-forward, or packet-switch subnet. Nearly all the wide area networks (except those using satellites) have stored-and-forward subnets. When the packets are small and all the same size, they are often called cells.

When a point-to-point subnet is used, an important design issue is what the router interconnection topology should look like. Figure 12-2 shows several possible topologies. Local networks that are designed as such usually have a symmetric topology. In contrast, wide area networks typically have irregular topologies.

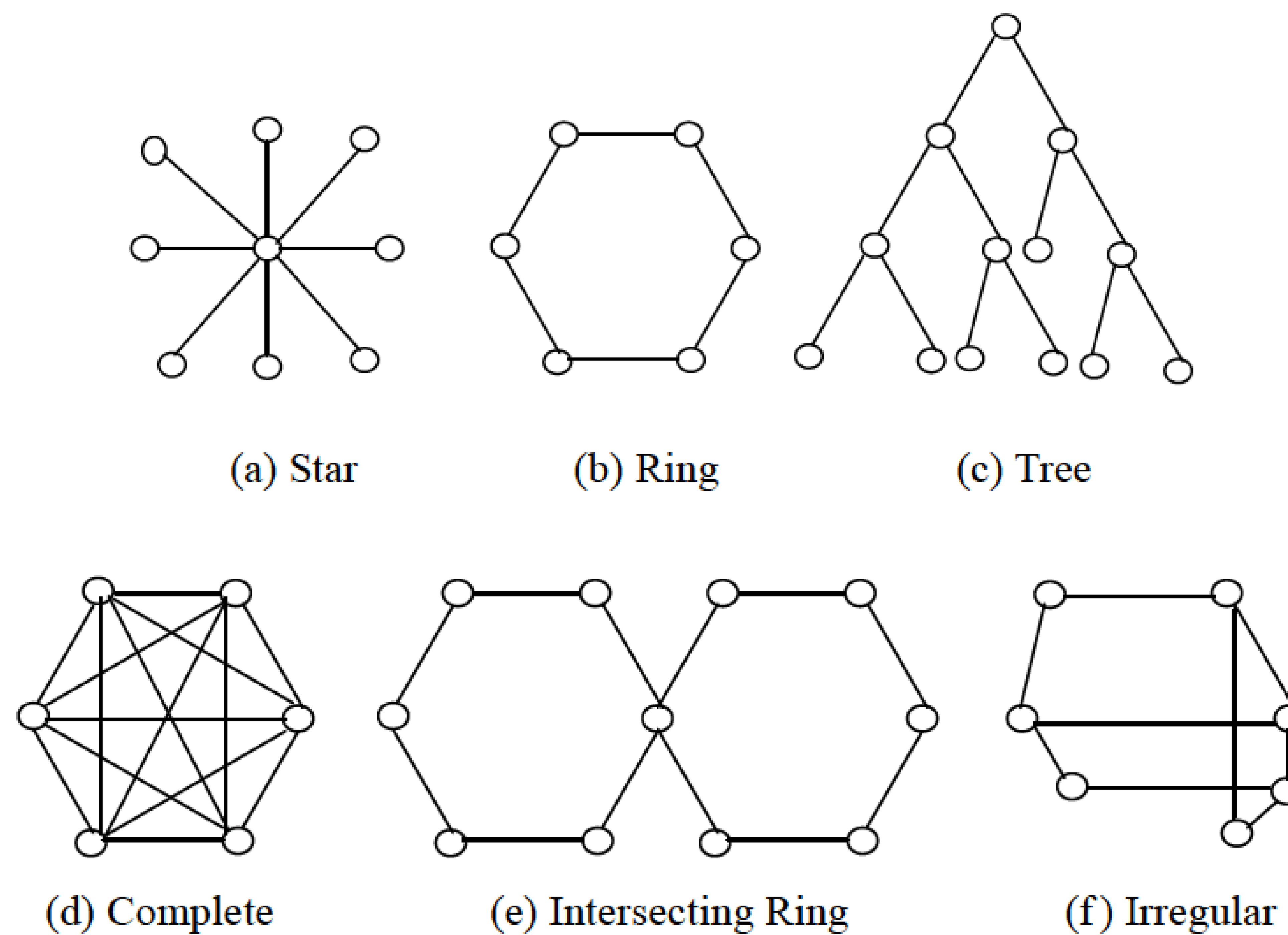


Figure 12-2 Some Possible Topologies for A Point-To-Point Subnet

12.3.2 Narrowband ISDN

A good example of a circuit-switched network is the Integrated Services Digital Network (ISDN), an evolving communications network standard that provides universal end-to-end connectivity over digital lines. ISDN provides data rates of 64 Kbps to 2.048 Mbps and we refer to it as N-ISDN (Narrowband ISDN), to contrast it with broadband ISDN (ATM) to be discussed later. N-ISDN was a massive attempt to replace the analog telephone system with a digital one suitable for both voice and non-voice traffic.

The purpose of the ISDN is to provide fully integrated digital services to users. These services fall into three categories: bearer services, teleservices, and supplementary services.

1. Bearer service

Bearer services provide the means to transfer information (voice, data, and video) between users without the network manipulating the content of that information. Bearer services belong to the first three layers of the OSI model and are well defined in the ISDN standard. They can be provided using circuit-switched, packet-switched, frame-relay, or cell-relay networks.

2. Teleservice

In teleservicing, the network may change or process the contents of the data. These services correspond to layers 4-7 of the OSI model.Teleservices rely on the facilities of the bearer services and are designed to accommodate complex user needs without the user having to be aware of the details of the process. Teleservices include telephony, teletex, telefax, videotex, telex, and teleconferencing. Although the ISDN defines these services by names, they have not yet become standards.

3. Supplementary Service

Supplementary Services are those services that provide additional functionality to the bearer services and teleservices. Examples of these services are reverse charging, call waiting, and message handling, all familiar from today's telephone company services.

12.3.3 X.25 Network

The packet switching protocol most widely used today is called X.25. Developed by the ITU-T in 1976, X.25 has been revised several times.

According to the formal definition given in the ITU-T, standard X.25 is an interface between data terminal equipment and data circuit terminating equipment for terminal operation at the packet mode on public data networks. Informally, we can say that X.25 is packet switching protocol used in a wide area network.

X.25 defines how a packet-mode terminal can be connected to a packet network for the exchange of data. It describes the procedures necessary for establishing, maintaining, and terminating connections (such as connection establishment, data exchange, acknowledgement, flow control, and data control). It also describes a set of services, called facilities, to provide functions such as reverse charge, call direct, and delay control.

Most X.25 networks work at speeds up to 64 kbps, which makes them obsolete for many purposes. Nevertheless, they are still widespread, so readers should be aware of their existence.

X.25 is connection-oriented and supports both switched virtual circuits and permanent ones. A switched virtual circuit is created when one computer sends a packet to the network asking to make a call to a remote computer. Once established, packets can be sent over the connection, always arriving in order. X.25 provides flow control, to make sure a fast sender cannot swamp a slow or busy receiver.

A permanent virtual circuit is used the same way as a switched one, but it is set up in advance by agreement between the customer and the carrier. It is always present, and no call setup is required to use it. It is analogous to a leased line.

12.3.4 Frame Relay

Frame relay is a service for people who want an absolute bare-bone connection-oriented way to move bits from A to B at reasonable speed and low cost. Frame relay can best be thought of as a virtual leased line. The customer leases a permanent virtual circuit between two points and can then send frames (i.e., packets) of up to 1600 bytes between them. It is also possible to lease permanent virtual circuits between a given site and multiple other sites, so each frame carries a 10-bit number telling which virtual circuit to use.

In addition to competing with leased lines, frame relay also competes with X.25 permanent virtual circuits, except that it operates at higher speeds, usually 1.5 Mbps, and provides fewer features.

12.3.5 Broadband ISDN and ATM

As application using the telecommunications networks advanced, however, the data rates of 64 Kbps to 2.048 Mbps, which narrowband ISDN provides, proved inadequate to support many applications. In addition, the original bandwidths proved too narrow to carry the large number of concurrent signals produced by a growing industry of digital service providers.

To provide for the needs of the next generation of technology, an extension of ISDN, called broadband ISDN (B-ISDN) is under study.

B-ISDN is a new wide area service. It will offer video on demand, live television from many sources, full motion multimedia electronic mail, CD-quality music, LAN interconnection, high-speed data transport for science and industry and many other services that have not yet even been thought of, all over the telephone line.

The underlying technology that makes B-ISDN possible is called ATM (Asynchronous Transfer Mode) because it is not synchronous (tied to a master clock), as most long distance telephone lines are. Note that the acronym ATM here has nothing to do with the Automated Teller Machines many banks provide (although an ATM machine can use an ATM network to talk to its bank).

The basic idea behind ATM is to transmit all information in small, fixed-size packets called cells. The cells are 53 bytes long, of which 5 bytes are header and 48 bytes are payload. ATM is both a technology (hidden from the users) and potentially a service (visible to the users). Sometimes the service is called cell relay, as an analogy to frame relay.

ATM networks are connection-oriented. Making a call requires first sending a message to set up the connection. After that, subsequent cells all follow the same path to the destination. Cell delivery is not guaranteed, but their order is. If cells 1 and 2 are sent in that order, then if both arrive, they will arrive in that order, never first 2 then 1.

ATM networks are organized like traditional WANs, with lines and switches (routers). The intended speeds for ATM networks are 155 Mbps and 622 Mbps, with the possibility of gigabit speeds later. The 155 Mbps speed was chosen because this is about what is needed to transmit high definition television. The exact choice of 155.52 Mbps was made for compatibility with AT&T's SONET transmission system. The 622 Mbps speed was chosen so four 155 Mbps channels could be sent over it. By now it should be clear why some of the gigabit testbeds operated at 622 Mbps: they used ATM.

Technical Notes to the Text

1. Ethernet，以太网。以太网是由美国 Xerox 公司、DEC 公司和 Intel 公司开发的一种局域数据通信网。

2. CDMA/CD，带冲突检测的载波侦听多路访问协议。该协议适合总线型网络，即以太网。

3. Timer，计时器，又称定时器。计时器指按一定的时间间隔来改变其内容用以度量时间的一种寄存器。

4. integrated services digital network (ISDN)，综合业务数字网。ISDN 将电话语音和计算机多媒体数据集成到一条高速的数字传输网络线路中，从而仅通过一条“线路”就可为客户同时提供语音服务和数据服务。N-ISDN (narrowband ISDN，窄带 ISDN)提供从 56Kb/s~2Mb/s 的低速服务。B-ISDN (broadband ISDN，宽带 ISDN)运用了 ATM 技术，可以提供 2~600Mb/s 的高速连接。根据 CCITT 的定义，B-ISDN 是指转移(传输、复用、交换)速率高于基群速率(2.048Mb/s 或 1.544Mb/s)的系统。

5. frame relay，帧中继。帧中继是一种新型的数据传输网络，之所以称为帧中继，是因为网上的操作大多基于 OSI 参考模型的第 2 层，即数据链路层，也称帧层。

6. ATM (asynchronous transfer mode)，异步传输模式。其基本速率是 150Mb/s，可支持高清晰度电视(HDTV)、多媒体会议电视、彩色传真、远端交互教育系统等宽带业务。

Word Bank to the Text

A. Useful new words

proximity	*n*. 接近，亲近
baseband	*n*. 基带，基本频带
notation	*n*. 符号
unregulated	*adj*. 无规定的，未调整的
collision	*n*. 碰撞，冲突
standardize	*v*. 使符合标准，使标准化
detection	*n*. 侦查，探测
robustness	*n*. 鲁棒性，坚固性，健壮性
propagate	*v*. 传播，宣传
placeholder	*n*. 占位符
circulate	*v*. 运行，循环
regenerate	*v*. 新生，重建
delimiter	*n*. 定界符，分隔符
interval	*n*. 间隔，距离
disrupt	*v*. 使中断，使分裂
comparable	*adj*. 可比较的，比得上的
provision	*n*. 限制性条款，附带条件
allotment	*n*. 分配
monopolize	*v*. 独占，垄断
decrement	*n*. 消耗
countdown	*n*. 倒数计秒
unidirectional	*adj*. 单向的，单向性的
downstream	*adv*. 顺流而下

entirety	*n*. 全部，完全
symmetric	*adj*. 相称性的，均衡的
universal	*adj*. 普遍的，通用的
accommodate	*v*. 供应，供给，使适应，调节
establishment	*n*. 确立，制定
absolute	*adj*. 完全的，绝对的
narrowband	*n*. 窄带
payload	*n*. 有效载荷

B. Useful expressions

in the proximity of	在……附近
collide into	与……相撞，冲突
have reservations about	对……有所保留
up to	多达
for one thing	一方面，首先
for another	另一方面，其次
hold up	阻挡，拦截
bring down	打倒，降低
take turns	轮流，依次
pass by	经过，掠过
replace with	取代，以……代替
impose on	施加影响于
be designated as	被指定为
run out	完成，被用完
equal to	等于，胜任
go into debt	负债
cover the expense	负担费用
contrast with	和……形成对照
define by name	通过名字来定义
in order	按顺序
in advance	事先，预先
compete with	比赛，竞争
video on demand	视频点播

C. Technical terms and proper names

Local Area Network (LAN)	局域网
Ethernet	以太网
baseband	基带
broadband	宽带

CSMA/CD	有冲突检测的载波侦听多路访问
token	令牌
token bus	令牌总线
token ring	令牌环
time register	时间寄存器
synchronous allocation (SA)	同步分配
target token rotation time (TTRT)	目标令牌循环时间
absolute maximum time (AMT)	绝对最大时间
timer	计时器
token rotation timer (TRT)	令牌循环计时器
token holding timer (THT)	令牌拥有计时器
DQDB (distributed queue dural bus)	分布式队列双总线
Wide Area Network (WAN)	广域网
point-to-point	点到点
store-and-forward	存储转发
circuit-switched	电路交换
packet-switch	分组交换
integrated services digital network (ISDN)	综合业务数字网
N-ISDN (narrowband ISDN)	窄带 ISDN
B-ISDN (broadband ISDN)	宽带 ISDN
bearer service	载体服务
teleservice	远程服务
supplementary service	附加服务
data terminal equipment	数据终端设备
data circuit terminating equipment	数据电路设备
connection-oriented	面向连接的
switched virtual circuit	交换式虚电路
permanent virtual circuit	永久式虚电路
frame relay	帧中继
ATM (asynchronous transfer mode)	异步传输模式
cell	信元
cell relay	信元中继

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. Networks with nodes that are in close physical proximity — within the same building, for instance, are called ____________ .

2. Whenever multiple users have unregulated access to a single line, there is a danger of signals overlapping and destroying each other. Such overlaps, which turn the signals into unusable noise, are called ____________ .

3. A ____________ is created when one computer sends a packet to the network asking to make a call to a remote computer.

4. ____________ is set up in advance by agreement between the customer and the carrier. It is always present, and no call setup is required to use it.

5. When the logical ring is initialized, the highest numbered station may send the first frame. After it is done, it passes permission to its immediate neighbor by sending the neighbor a special control frame called a ____________ .

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. In token ring, at the sending station is the token when a data frame is in circulation. (   )
2. Another term for CSMA/CD and the IEEE 802.3 standard is FDDI. (   )
3. In ISDN teleservices, the network can change or process the contents of the data. (   )
4. Frame relay can best be thought of as a virtual leased line. (   )
5. If two routers that do not share a cable, nevertheless, wish to communicate, they must do this indirectly, via other routers. (   )

Ⅲ. Match each of the following terms with the appropriate definition.

ISDN	SA	Bearer services

1. ________ register holds the amount of time a station has to send its synchronous data.
2. ________ an acronym for Integrated Services Digital Network.
3. ________ to provide the means to transfer information (voice, data, and video) between users without the network manipulating the content of that information.

Ⅳ. List four possible topologies for a point-to-point subnet.

1. ______________________________
2. ______________________________
3. ______________________________

4. ______________________________

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

circulate	proximity	comparable
regenerate	collision	monopolize
interval	standardize	universal

1. They are controlling so much cocoa that they are virtually ________ the market.
2. Rumours were already beginning to ________ that the project might have to be abandoned.
3. Farmers were meant to get an income ________ to that of town's people.
4. Nerve cells have limited ability to ________ if destroyed.
5. He feels ________ education does not benefit those children who are either below or above average intelligence.
6. The insurance industry has produced its own proposals for ________ health care.
7. The Chinese seafood restaurant benefits from its ________ to several cinemas.
8. The ferry service between Burnham and Wallasea Island has restarted after an ________ of 12 years.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can **"transmit"** and fill in the sentences with the right word.

Example: In the CSMA/CD the station wishing to **transmit** first **listens** to make certain the link is free, then **transmits** its **data**, then listens again.

1. Some di______ are transmitted from one generation to the next.
2. The ce______ was transmitted live by satellite to over fifty countries.
3. According to the author, one of the important yet neglected functions of institutions of higher education is preparing their students to transmit inherited kn______ .
4. General education is designed to transmit a common cultural he______ rather than to develop trained specialists.
5. Fibre-optic cables will transmit the electronic si______ .
6. Now people can send telegrams, telephone each other, and transmit in______ much more quickly than ever before.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

A local-area network is a computer network __1__ a small geographic area, like a home, office, or a school. The defining characteristics of LANs, in contrast __2__ wide-area networks

(WANs), include their much higher data-transfer ___3___, smaller geographic range, and lack of a need for leased telecommunication lines. LANs may have connections with other LANs via leased lines or services. Depending on how the connections are made, secured, and the distance ___4___ they become a Metropolitan Area Network (MAN), a Wide Area Network (WAN), or a part of the Internet.

A MAN is optimized ___5___ a larger geographical area than a LAN, ___6___ from several blocks of buildings to entire cities. A MAN might be owned and ___7___ by a single organization, but it usually will be used by many individuals and organizations. MANs can ___8___ up to 50km, devices used are modem and wire/cable WANs span broad geographical ___9___. The most universal and powerful WAN is the Internet. MANs' geographic scope falls ___10___ a WAN and LAN. MANs provide Internet connectivity for LANs in a metropolitan region, and connect them to wider area networks like the Internet.

1. A. covering	B. cover	C. covered	D. to cover
2. A. with	B. to	C. as	D. in
3. A. speeds	B. ratios	C. bits	D. rates
4. A. to be involved	B. involves	C. involved	D. involving
5. A. for	B. as	C. within	D. with
6. A. ranges	B. ranging	C. ranged	D. to range
7. A. bought	B. managed	C. operated	D. borrowed
8. A. speed	B. take	C. extend	D. span
9. A. areas	B. distances	C. zips	D. zones
10. A. behind	B. between	C. into	D. through

Translation

Ⅷ. Translate the following into Chinese.

1. Whenever multiple users have unregulated access to a single line, there is a danger of signals overlapping and destroying each other.

2. Through this mechanism, the station holding the reservation gets the opportunity to transmit as soon as the token is free, whether or not it comes next physically on the ring.

3. If two routers that do not share a cable, nevertheless, wish to communicate, they must do this indirectly, via other routers.

4. Examples of these services are reverse charging, call waiting, and message handling, all familiar from today's telephone company services.

5. The underlying technology that makes B-ISDN possible is called ATM (Asynchronous Transfer Mode) because it is not synchronous (tied to a master clock), as most long distance telephone lines are.

Chapter 13
Internet

Pre-reading Questions

1. What are domain and DNS?
2. How many services are provided by the Internet? Name each of them.
3. Identify two ways accessing to Internet, and explain the differences between them.

13.1 Introduction

The Internet is a worldwide and opening interconnected computer network. It is a special network, on which users can administer themselves. Internet is based on TCP/IP protocol. It can provide various information resources and many network services for network users. Basic functions of Internet include: sending and receiving E-mail, issuing news, Telnet, file transfer, information access, etc. Internet has already become the widest influential international linking network with the largest scale and the most users. A network is a collection of computers that are connected together so they can share information, and the Internet is a network of networks.

The Internet, or Net, was launched in 1969 when the United States funded a project and developed a national computer network called Advanced Research Project Agency Network (ARPANET). The Internet is a large network that connects together smaller networks all over the globe. The Web, also known as WWW and the World Wide Web, was introduced in 1992 at the Center for European Nuclear Research (CENR) in Switzerland. Prior to the Web, the Internet was all text, no graphics, animations, sound, or video. The Web made it possible to include these elements. It provided a multimedia interface to resources available on the Internet. From these early research beginnings, the Internet and the Web have evolved into one of the most powerful tools of the 21st century.

It is easy to get the Internet and the Web confused, but they are not the same thing. The Internet is the actual physical network that is made up of wires, cables, and satellites. Being connected to this network is often described as being online. The Internet connects millions of computers and resources throughout the world. The Web is a multimedia interface to resources available on the Internet. Every day over a billion users from every country in the world use the Internet and the Web.

13.2 Technology of Internet

13.2.1 Internet Address

The Internet is made up of hundreds of connected computers and machines. Every machine on the Internet has a unique identifying number, called an IP Address. A typical IP address looks like this:

216.27.61.137

To make it easier for us to remember, IP addresses are normally expressed in decimal format as a “dotted decimal number” like the one above. But computers communicate in binary form. Look at the same IP address in binary:

11011000.00011011.00111101.10001001

According to the scale of the network, IP address can be divided into five classes: class A, B, C, D, E, respectively. Table 13-1 illustrates some typical examples of the five classes respectively.

Table 13-1 examples of the five classes of IP address

Class	Net	Host or Node
A	115.	24.53.107
B	145.24.	53.107
C	195.24.53.	107
D	224.	24.53.107
E	232.	24.53.107

Class A — This class is for very large networks, such as a major international company might have. IP addresses with a first octet from 1 to 126 are part of this class. The other three octets are used to identify each host (Table 13-1). This means that there are 126 Class A networks each with 16777214 (2^{24} - 2) possible hosts for a total of 2147483648 (2^{31}) unique IP addresses. Class A networks account for half of the total available IP addresses. In Class A networks, the high order bit value (the very first binary number) in the first octet is always 0.

Loopback — The IP address 127.0.0.1 is used as the loopback address. This means that it is used by the host computer to send a message back to itself. It is commonly used for troubleshooting and network testing.

Class B — Class B is used for medium-sized networks. A good example is a large college campus. IP addresses with a first octet from 128 to 191 are part of this class. Class B addresses also include the second octet as part of the Net identifier. The other two octets are used to identify each host (Table 13-1). This means that there are 16384 (2^{14}) Class B networks each with 65534 (2^{16} - 2) possible hosts for a total of 1073741824 (2^{30}) unique IP addresses. Class B networks make up a quarter of the total available IP addresses. Class B networks have a first bit value of 1 and a second bit value of 0 in the first octet.

Class C — Class C addresses are commonly used for small to mid-size businesses. IP

addresses with a first octet from 192 to 223 are part of this class. Class C addresses also include the second and third octets as part of the Net identifier. The last octet is used to identify each host (Table 13-1). This means that there are 2097152 (2^{21}) Class C networks each with 254 ($2^8 - 2$) possible hosts for a total of 536870912 (2^{29}) unique IP addresses. Class C networks make up an eighth of the total available IP addresses. Class C networks have a first bit value of 1, second bit value of 1 and a third bit value of 0 in the first octet.

Class D — Used for multicasts, Class D is slightly different from the first three classes. It has a first bit value of 1, second bit value of 1, third bit value of 1 and fourth bit value of 0. The other 28 bits are used to identify the group of computers the multicast message is intended for (Table 13-1). Class D accounts for 1/16th (268435456 or 2^{28}) of the available IP addresses.

Class E — Class E is used for experimental purposes only. Like Class D, it is different from the first three classes. It has a first bit value of 1, second bit value of 1, third bit value of 1 and fourth bit value of 1. The other 28 bits are used to identify the group of computers the multicast message is intended for (Table 13-1). Class E accounts for 1/16th (268435456 or 2^{28}) of the available IP addresses.

Broadcast — Messages that are intended for all computers on a network are sent as broadcasts. These messages always use the IP address 255.255.255.255.

IP Addresses are used whenever you use E-mail, whenever you connect to the World Wide Web (WWW), whenever you communicate to someone else on a different computer.

Some computers have permanent IP Addresses, while others do not. For example, computers that are permanently connected to the Internet, like servers (i.e. Yahoo, etc.) usually have permanent IP Addresses — that is, they are given one, and only one IP Address. On the other hand, some computers (usually home computers connecting to the Internet via a modem or server) can have different IP Addresses for every time that they log in. You might get one IP Address when you connect to the Internet one day, and then a different address the next day!

13.2.2　DNS (Domain Name System)

1. Domain Name

The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme. It is primarily used for mapping host names and E-mail destinations to IP addresses but can also be used for other purposes. DNS is defined in RFCs 1034 and 1035.

Very briefly, the way DNS is used as follows. To map a name onto an IP address, an application program calls a library procedure called the resolver, passing it the name as a parameter. The resolver sends a UDP packet to a local DNS server, which then looks up the name and returns the IP address to the resolver, which then returns it to the caller. Armed with the IP address, the program can then establish a TCP connection with the destination, or send it UDP packets.

When you want to access a website on the Internet, you have to type in the website's URL, or

Uniform Resource Locator. This URL is the specific address of the website, and thus lets your computer find that site. A URL might look like:

http://www.yahoo.com/C2006/

The "www.yahoo.com" part of the URL is the domain name. Domain names tell the computer where to look for the server of the website. Domain names can refer to specific computers, organizations, companies, or other entities. Each domain is named by the way upward from it to the (unnamed) root. The components are separated by periods (pronounced "dot"). Thus Sun Microsystems engineering department might be eng.sun.com, rather than a UNIX-style name such as /com/sun/eng.

Domain names are case insensitive, so edu and EDU mean the same thing. Component names can be up to 63 characters long, and full path names must not exceed 255 characters.

There are four levels of domain names:

(1) Top-Level Domains

The top-level domain name is the .com part of "www.yahoo.com". The top-level domain names refer to the type of organization that is acting as the server for the website:

- .com — Websites for commercial purposes
- .edu — Educational institutes
- .net — Internet-related websites
- .org — Non-profit organizations
- .gov — government websites
- .mil — military websites

In addition to the top level domain names listed above, every country in the world has its own unique top-level domain name. For example:

- .au — Australia
- .br — Brazil
- .ca — Canada
- .cn — China
- .uk — United Kingdom
- .us — United States

The country domains include one entry for every country, as defined in ISO 3166.

(2) Second-Level Domains

The second-level domain name would be the yahoo.com part of "www.yahoo.com". Second-level domains can be registered by users or other organizations. For example, Yahoo registered the domain name yahoo.com, while Microsoft registered the domain name, microsoft.com.

Second-level domain names can have up to 61 characters. They must be unique to the organization that registered it. Two organizations cannot have the same second-level domain name.

(3) Third-Level Domains

The third-level domain name would be the whole "www.yahoo.com". It usually specifies the company hosting the domain. Usually, it is "www.".

2. Domain Name Server

As you know, every single computer that is connected to the Internet has a unique IP Address. This address identifies the specific computer for all of the other computers connected to the Internet.

But who could possibly remember a long string of numbers? Names like www.yahoo.com, www.amazon.com, and www.google.com are so much easier to remember! Obviously, it is much easier to remember domain name. All domain names have a corresponding IP Address, so domain names must be translated to an IP Address in order for computers to be able to communicate and exchange information.

When you first get on the Internet, and you type in the URL: http://www.yahoo.com/C2006/, the first thing that happens is that the DNS will change "www.yahoo.com" into the numbers of the IP Address. Then, the file will be transferred from the Yahoo server to your computer for you to view!

13.2.3　HTTP and SMTP

1. HTTP (Hyper Text Transfer Protocol)

The standard Web transfer protocol is HTTP. Each interaction consists of one ASCII request, followed by one RFC 822 MIME-like response. Although the use of TCP for the transport connection is very common, it is not formally required by the standard. HTTP is constantly evolving. Several versions are in use and others are under development.

The HTTP protocol consists of two fairly distinct items: the set of requests from browsers to servers and the set of responses going back the other way. And all the newer versions of HTTP support two kinds of requests: simple requests and full requests.

2. SMTP (Simple Mail Transfer Protocol)

Within the Internet, E-mail is delivered by having the source machine establish a TCP connection to port 25 of the destination machine. Listening to this port is an E-mail daemon that speaks SMTP. This daemon accepts incoming connections and copies messages from them into the appropriate mailboxes. If a message cannot be delivered, an error report containing the first part of the undeliverable message is returned to the sender.

SMTP is a simple ASCII protocol. After establishing the TCP connection to port 25, the sending machine, operating as the client, waits for the receiving machine, operating as the server, to talk first. The server starts by sending a line of text giving its identity and telling whether or not it is prepared to receive mail. If it is not, the client releases the connection and tries again later.

Even though the SMTP protocol is well defined (by RFC 821), a few problems can still arise. One problem relates to message length. Some older implementations cannot handle messages exceeding 64KB. Another problem relates to timeouts. If the client and server have different timeouts, one of them may give up while the other is still busy, unexpectedly terminating the connection. Finally, in rare situations, infinite mail-storms can be triggered. For example, if host 1

holds mailing list A and host 2 holds mailing list B and each list contains an entry for the other one, then any message sent to either list will generate a never-ending amount of E-mail traffic.

13.3 Services Provided by the Internet

The software that supports the Internet provides a large number of technical services upon which everything else is built. Most of these services operate behind the scenes, and you do not need to understand them. Nevertheless, there are four important Internet services we do need to talk about. You don't need to know the details, but you need to know they exist.

13.3.1 WWW (World Wide Web)

WWW is the section of the Internet that contains all of the web pages.

One of the most common reasons why people use the WWW is to do research. The World Wide Web holds thousands of web pages created by all sorts of people. People usually create informative web sites about the topics they are interested in.

Another reason for accessing the World Wide Web is for commercial purposes. As the Internet and the WWW has been gaining popularity, businesses have found it extremely beneficial to create web sites. On many of these websites, people can purchase items online, and find information about the store, and people can find information about the products and prices! Now, there are even on-line auctions!

The World Wide Web is also often used for entertainment. Thousands of people go online to play games! In addition to games, the WWW gives people access to all sorts of music and other multimedia devices!

Finally, the World Wide Web can also be used for personal reasons. Many people put up personal websites about themselves and their interests. They create personal sites for fun and to become more involved in the comings and goings of the Internet!

As you might know, all web pages have a unique address. When you type in this address at the top of your Internet browser, the designated site will pop up in the Internet browser. These addresses are called URLs, or Uniform Resource Locators.

A web browser is a tool that lets you view web pages, just as word processors let you view text documents. Web browsers have now evolved to contain tons of special features and tools to make your life easier! Today, the two most commonly used web browsers are Microsoft Internet Explorer and Netscape Navigator.

13.3.2 E-mail

Frequent Internet users often consider regular mail as "snail mail. It's so much slower than E-mail, or electronic mail! With regular mail, it can take days for someone to receive your letter. However, with E-mail, the message gets sent instantly — well, almost instantly! Best of all, E-mail

doesn't have cost that regular mail has!

People all over the world sent E-mail to each other. You can E-mail your family, your co-workers, and even your senator. All you need to send and receive E-mail is an E-mail account, access to the Internet, and an E-mail program.

A typical E-mail message has two basic elements:

- Header: The header of an E-mail message contains the name and the E-mail address of the recipient of the message, as well as the message's subject. If you are sending the message to more than one person, the header will contain the names and addresses of all the people you are sending it to.
- Body: The body of an E-mail message is the message itself.

Many E-mail programs allow you to attach files such as documents and worksheets. If a message has an attachment, the file name appears on the attachment line.

How Does E-mail Work?

First, let's take a look at an E-mail address. A typical E-mail address might look like:

username@mail.com

On the left side of the axon sign (@) is the user name. The user name acts like a mailbox (more on this later). On the right side of the axon sign is the domain name. In a nutshell, the domain name in an E-mail address is the name of the E-mail server the user is using.

When you write a letter to your friend, you type up the message, type in her E-mail address, and click the little "send" button. The E-mail gets sent from one computer to another via computers known as mail servers. The message will travel from one mail server to another, until it finally reaches its destination mail server. There, it will be stored in the recipient's "mailbox", which is determined by his or her user name, until he or she picks it up.

13.3.3　FTP (File Transfer Protocol)

FTP is an Internet standard for transferring files. Many computers on the Internet allow you to copy files from one computer to another.

You can get files from other computers and put them on your computer. This process is called downloading. You can put your files onto other people's computers. This process is called uploading. You can move files between your Internet accounts.

There are two main ways to use FTP.

One is through the World Wide Web. This method allows you to use FTP through your World Wide Web browser. It is probably the most convenient way to search for and get publicly available files, but it is somewhat limited. It doesn't handle large files efficiently, or allow you to put files onto other people's computers or move files around.

The other is using specific FTP software. This is a more powerful and flexible way to search for and get files from the Internet. You can perform the full range of FTP functions using this method — this includes putting a file from your computer onto another computer.

13.3.4 Telnet

Many computers on the Internet will allow you to connect to them and to run selected programs on them. Telnet is an Internet standard that allows you to connect to another computer (host) on the Internet and to log on to that computer as if you were on a terminal in the next room. There are hundreds of computers on the Internet that you can connect to. Some allow limited free access, and others charge fees for their use.

13.4 Networking Devices

13.4.1 Hub and Repeater

1. Repeater

Different types of network cabling have their own maximum distance that they can move a data signal. In cases where a LAN must be extended beyond its maximum run for its particular cabling type, repeaters are used. Repeaters take the signal that it receives from the computers and other devices on the LAN and regenerates the signal in order for it to maintain its integrity while traveling along a longer media run that is normally possible.

Repeaters do not have any capability of directing network traffic or deciding what particular route that certain data should take, they are simply devices that sit on the network and boost the data signal that they receive. The problem with repeaters is that they amplify the entire signal that they receive, including any line noise. In the worst case scenario, they pass on data traffic that is barely discernable from the background noise on the line.

2. Hub

Hubs organize your cables and relay data signals to all computers that exist on your LAN.

Hubs are used on networks where twisted-pair cabling is used. The ports, which are available on the hub, provide connection points for the devices on the network. Computers and devices are connected to the hub via network cables to individual ports. In cases where a LAN outgrows the size of its hub, a new hub can be attached by daisy-chaining them together using a short connection cable.

13.4.2 Bridge and Switch

1. Bridge

The bridges used in computer networking are not like your typical bridge. This type of bridge is an internetworking device used to help conserve the bandwidth on the network. When LANs are really starting to grow, network data traffic begins to become a little overwhelming to the available bandwidth on the network media.

One way of conserving this network traffic is to slice the network up into smaller segments.

These segments are connected to a bridge. Bridges are smarter than hubs and repeaters. They also use software to help get the job done. Bridges are able to read MAC addresses, which are also known as hardware addresses that are burned onto the NIC inside every computer on the network. It is by learning which MAC addresses are live on the network that the bridge is able to control traffic that is local to a particular segment and keep it from spreading to other network segments that are being serviced by another bridge switch.

2. Switch

A switch is another internetworking device used to manage bandwidth on a large network. Switches are rapidly becoming one of the most used internetworking devices from connecting even smaller networks because they allow you to have some control over the use of the bandwidth on the network. A switch controls how data flows by using the MAC addresses that are placed on each data packet. This MAC address on the data packet is the same MAC address on the network card of that particular computer.

Switches divide networks into what is known as a Virtual LAN or VLAN. The best thing about a VLAN, which is also a logical grouping of computers on a network into what is described as some sort of communication group, is that the computers really do not have to be in close proximity or even on the same floor. This is because it allows the computers to be grouped by the similarities in the types of users in the VLAN.

Switches use a combination of software and hardware to switch packets between computers and other devices on the network. When the computers are directly connected to the switch, the switch can supply each of the computers with a dedicated amount of bandwidth. For example, say users are on a 100Mbps Ethernet network via a switch. Each user can realize a bandwidth of 100Mbps and don't have to compete for the bandwidth the way computers do on a network via a hub because each port on the switch has a dedicated 100Mbps. This is why switches are rapidly replacing hubs. Inexpensive switches are even available to accommodate small networks and home network markets.

Some of the switch hardware can also take advantage of full-duplex access to the network media, which allows for the sending and receiving of data simultaneously.

Since switches are becoming very popular on both the small and large networks, they have all but replaced bridges as the internetworking devices for conserving network bandwidth and expanding LANs into larger corporate internetwork. But, they are also making hubs a device of the past for smaller networks.

13.4.3 Router

At first, let us comprehend packets. As you know, the file that you want to access (for example, an E-mail message, a graphics file, an HTML file, etc.) is broken up several small chunks of data. These chunks, called packets, are then each assigned a separate number, and routed to your computer (or whichever computer the user specified it to go to). The packets don't all have to take the same route to get to the designated location. They could all take different routes, and then when all the

packets arrive, the packets are combined into the file's original form, all ready for you to access!

A packet is typically 64 kilobyte in size. Sometimes, if the file is really large, there can be thousands of packets! Each of these packets will make their way individually to the destination!

The concept of packets and packet-switching networks (networks making use of the idea of packets) are widely used throughout the Internet, and are extremely efficient!

Routers are the devices that make the Internet possible. Routers are connected to two or more networks at crossing points. When they receive a packet of information, they decide which way they should send the packet to ensure that it gets to its destination quickly and efficiently. For example, say you want to send an E-mail to your friend. The E-mail message will first be broken up into small packets. Each packet will contain the IP Address of the destination computer. Then, your computer will ship all the packets to the nearest router, which will be connected to many networks. The router then decides which network it should send the packet to in order to best reach your friend's computer!

13.5 Access to Internet

13.5.1 Dial-up

Refering to connecting a device to a network via a modem and a public telephone network, dial-up access is really just like a phone connection, except that the parties at the two ends are computer devices rather than people. Because dial-up access uses normal telephone lines, the quality of the connection is not always good and data rates are limited. In the past, the maximum data rate with dial-up access was only 56 Kbps (56,000 bits per second).

An alternative way to connect two computers is through a leased line, which is a permanent connection between two devices. Leased lines provide faster throughput and better quality connections, but they are also more expensive.

13.5.2 DSL

Digital Subscriber Line, or DSL, is a method of accessing the internet via telephone lines. It differs from dial-up access in that it makes use of a special frequency spectrum which allows both digital information and telephone services to run through the same cable. It is also a good deal faster than dial-up Internet, theoretically processing upwards of 10 megabits per second, in contrast to dial-up's sluggish 56 kilobits (or roughly five percent of a megabit). In fact such extremely fast speeds are rarely obtained by DSL services because most phone companies install artificial "speed caps" to impose a universal level of service on their lines. However, even a slower-paced DSL connection can make for easy and responsive web navigation.

A DSL connection requires use of a specialized modem which converts the incoming frequencies passed through a phone line into the digital information used by computers, and reverses the process for outgoing transmissions. To avoid household telephone equipment interfering with the DSL signal, a special filtration device must be installed at all telephone jacks

in the home. Such filters are often included with the purchase of a DSL modem and are commonly supplied by one’s internet service provider as well.

Technical Notes to the Text

1. IP address，IP 地址。为了通信方便，给每一台计算机都事先分配一个标识地址，该标识地址就是 IP 地址。根据 TCP/IP 协议规定，IP 地址由 32 位二进制数组成，而且在 Internet 范围内是唯一的，如 11010010.01001001.10001100.00000010。为了方便记忆，就将组成计算机的 IP 地址的 32 位二进制分成 4 段，每段 8 位，中间用小数点隔开，然后将每 8 位二进制转换成十进制数，上述计算机的 IP 地址就变成了 210.73.140.2。

2. Broadcast，广播。广播是指向这个网络内的所有结点发送数据。通常主机地址为全 1。

3. DNS (Domain Name System)，域名系统。IP 地址和域名都能表示一台连到 Internet 上的计算机，而把域名翻译成 IP 地址的工作由域名服务器完成，在 Internet 上有许多域名服务器负责翻译地址。

4. HTTP (hyper text transfer protocol)，超文本传输协议。该协议是为分布式超媒体信息系统设计的面向对象的传输协议。

5. SMTP (simple mail transfer protocol)，简单邮件传输协议。该协议是描述客户机与远程主机之间传送电子邮件的协议。

6. port，端口。端口是在计算技术和通信技术中的网点上的一种功能部件，通过它，数据可以进行通信。

7. WWW (World Wide Web)，万维网。万维网是指以超文本技术为基础，通过将位于 Internet 上不同地点的相关数据信息有机地编织在一起。WWW 是一个多媒体信息发布系统。

8. E-mail，电子邮件。电子邮件是 Internet 上使用最广泛和最受欢迎的服务之一，它是网络用户之间进行快速、简便、可靠且低成本联络的现代通信手段。

9. FTP (file transfer protocol)，文件传输协议。该协议是指在网络上的计算机之间传输文件的协议。用户可以将本地计算机上的文件传送到远程主机上(upload，称为上传)，也可从远程主机上获取所需要的文件(download，称为下载)。

10. Telnet，远程登录。在网络通信协议 telnet 的支持下，使用户的计算机成为某一远程计算机的仿真终端，共享远程计算机的软硬件资源及数据库和 Internet 的其他信息服务。

11. Dial-up，拨号。拨号是通过调制解调器或者公用电话网连接到互联网的一种方式。

12. DSL (digital subscriber line)，数字用户线。数字用户线是一种通过电话线访问互联网的方法。

Word Bank to the Text

A. Useful new words

administer　　*v.* 管理，执行

influential	*adj*. 有影响的，有势力的
unique	*adj*. 唯一的，独特的
respectively	*adv*. 分别地，各个地
octet=byte	*n*. 八位位组，八位字节
troubleshooting	*n*. 问题解答
identifier	*n*. 标识符
uniform	*adj*. 统一的，一致的
insensitive	*adj*. 对……没有感觉的，感觉迟钝的
undeliverable	*adj*. 无法投递的，无法送达的
trigger	*v*. 引起，触发
beneficial	*adj*. 有益的，受益的
auction	*n*. 拍卖
axon	*n*. 轴突
nutshell	*n*. 坚果壳
integrity	*n*. 完整，完整性
boost	*v*. 推进
amplify	*v*. 放大，增强
discernable=discernible	*adj*. 可辨别的，可认识的
twist	*v*. 扭曲，缠绕
outgrow	*v*. 过大而不适于，年久丧失(某种习惯、兴趣等)
conserve	*v*. 保存，保藏
overwhelming	*adj*. 压倒性的，无法抵抗的
dedicated	*adj*. 专用的
comprehend	*v*. 领会，理解
filtration	*n*. 过滤，筛选

B. Useful expressions

account for	说明，占
evolve into	发展[进化]成
be made up of	由……组成
log in (=log on)	进入系统
look up	寻访，探访
armed with	装备有……的
give up	放弃
for commercial purpose	为了商业目的
give access to	准许出入
pop up	突然出现

have capability of doing	有能力做……
pass on	传递
slice up into	切成一片片
keep from	阻止，抑制
in close proximity	在附近，邻近
compete for	为……竞争
take advantage of	利用
allow for	虑及，体谅
except that	除了……之外
make for	走向，有利于，有助于
interfere with	妨碍，干扰

C. Technical terms and proper names

access	访问，存取
Internet address	网络地址，IP 地址
domain	域
DNS (domain name system)	域名系统
URL (uniform resource locator)	统一资源定位器
Top-level domain	顶级域名
Second-level domain	二级域名
Third-level domain	三级域名
HTTP (hyper text transfer protocol)	超文本传输协议
SMTP (simple mail transfer protocol)	简单邮件传输协议
E-mail	电子邮件
E-mail daemon	电子邮件幽灵程序
FTP (file transfer protocol)	文件传输协议
download	下载
upload	上传
Telnet	远程登录
hub	集线器
port	端口
repeater	中继器
bridge	网桥
switch	交换机
router	路由器
packet	数据包

dial-up 拨号

DSL (digital subscriber line) 数字用户线

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. Internet is a worldwide and ________ interconnected computer network.
2. DNS is primarily used for mapping host names and E-mail destinations to ________.
3. On the Internet, the word address always refers to an ________ address, not a postal address.
4. SMTP is a simple ________ protocol.
5. ________ is the standard Web transfer protocol.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The HTTP protocol consists of two fairly distinct items. ( )
2. SMTP is a simple ASCII protocol. ( )
3. Telnet is an Internet standard that allows you to connect to another computer (host) on the Internet and to log on to that computer as if you were on a terminal in the next room. ( )
4. Digital Subscriber Line differs from dial-up access in that it makes use of a special frequency spectrum which allows both digital information and telephone services to run through the same cable. ( )
5. The router used in computer networking is an internetworking device used to help conserve the bandwidth on the network. ( )

Ⅲ. Match each of the following terms with the appropriate definition.

hub	switch	router

1. ________ An internetwork device that organizes your cables and relay data signals to all computers that exist on your LAN.
2. ________ An internetwork device that used to manage bandwidth on a large network. It controls how data flows by using the MAC addresses that are placed on each data packet.
3. ________ An internetwork device that connects to two or more networks at crossing points, decides which way the packet should send to, and ensure that the packet gets to its destination quickly and efficiently.

Ⅳ. Answer the following questions.

1. What is the Internet?
2. What are the uses of the Internet?
3. What are four important services provided by the Internet?
4. What is the essence of DNS?
5. List the common internetwork devices.

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

integrity	respectively	conserve
boost	uniform	comprehend
infinite	influential	amplify

1. Good communication skill helps him to have ________ friends.
2. They each excel in their ________ fields.
3. All flowing water, though it appears to be ________, is actually divided into extensive inner surfaces, or layers, moving against one another.
4. Separatist movements are a threat to the ________ of the nation.
5. The move is designed to ________ sales during the peak booking months of January and February.
6. Her anxiety about the world was ________ her personal fears about her future.
7. The republic's factories have closed for the weekend to ________ energy.
8. I cannot ________ how you could have been so stupid.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "identify" and fill in the sentences with the right word.

Example: IP addresses with a first octet from 1 to 126 are part of this class. The other three octets are used to **identify** each **host**.

1. Scientists claim to have identified ch______ produced by certain plants which have powerful cancer-combating properties.
2. Having identified the pr______, the question arises of how to overcome it.
3. Can you identify your um_______ among this lot?
4. The young man tried to identify her pe_______.
5. He was confident enough to identify be_______ with goodness.
6. The local police authorities have already identified around 10 murder su______.

Summary of the Text

Ⅶ. Choose the best one of the four Choices given to fill in each blank.

Internet is a worldwide and opening interconnected computer network. It is a special network, __1__ which users can administer themselves. Internet is based on TCP/IP __2__. It can provide various information resources and many network services for network __3__.

Intranet is the generic __4__ for a collection of private computer networks within an organization. An intranet uses network technologies as a tool to facilitate communication __5__ people or workgroups to improve the data __6__ capability and overall knowledge base of an organization's employees.

An Extranet is actually an Intranet that is partially __7__ to authorized outsiders. Intranet network is connected to the __8__, if it allows access __9__ the Internet, will be an Extranet. The actual server will reside behind a firewall. The firewall helps to control access between the Intranet and Internet __10__ access to the Intranet only to people who are suitably authorized.

1. A. on	B. in	C. for	D. with
2. A. agreement	B. serial	C. task	D. protocol
3. A. customers	B. consumers	C. computers	D. users
4. A. word	B. concept	C. term	D. name
5. A. among	B. between	C. in	D. at
6. A. share	B. sharing	C. shared	D. to share
7. A. accessible	B. access	C. accessibly	D. assess
8. A. intranet	B. extranet	C. internet	D. wannet
9. A. from	B. into	C. to	D. by
10. A. to permit	B. permit	C. permitted	D. permitting

Translation

Ⅷ. Translate the following into Chinese.

1. The ARPAnet was designed so that even if part of its physical structure were destroyed, information could still be sent to any remaining destination.

2. To map a name onto an IP address, an application program calls a library procedure called the resolver, passing it the name as a parameter.

3. Within the Internet, E-mail is delivered by having the source machine establish a TCP connection to port 25 of the destination machine.

4. An intranet uses network technologies as a tool to facilitate communication between people or workgroups to improve the data sharing capability and overall knowledge base of an organization's employees.

5. An organization's intranet typically includes Internet access but is firewalled so that its computers cannot be reached directly from the outside.

Chapter 14
Information Security

Pre-reading Questions

1. What is information security?
2. How to cryptograph? What is Cryptographic Principles?
3. What is network security technology? Name some of them.

For the first few decades of their existence, computer networks were primarily used by university researchers for sending E-mail, and by corporate employees for sharing printers. Under these conditions, security did not get a lot of attention. But now, as millions of ordinary citizens are using networks for banking, shopping, and filing their tax returns, information security is looming on the horizon as a potentially massive problem.

14.1 A Brief Overview of Information Security Concepts

Information Security Components: whereas information is transmitted, stored, encrypted or processed, its value derives from three main attributes or qualities, ie, Confidentiality, Integrity and Availability (CIA). Information Systems are decomposed in three main portions, hardware, software and communications with the purpose to identify and apply information security industry standards, as mechanisms of protection and prevention, at three levels or layers: Physical, personal and organizational.

Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction. The terms information security, computer security and information assurance are frequently used interchangeably. These fields are interrelated share the common goals of protecting the confidentiality, integrity and availability of information; however, there are some subtle differences between them. These differences lie primarily in the approach to the subject, the methodologies used，and the areas of concentration. Information security is concerned with the confidentiality, integrity and availability of data regardless of the form the data may take: electronic, print, or other forms.

Governments, military, financial institutions, hospitals, and private businesses amass a great deal of confidential information about their employees, customers, products, research, and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers. Should confidential information

about a businesses customers or finances or new product line fall into the hands of a competitor, such a breach of security could lead to lost business, law suits or even bankruptcy of the business.

Information security is concerned with every area，such as data and communications security. Nowadays, as millions of ordinary citizens are using networks for banking and shopping, network security is looming on the horizon as a potentially massive problem. In the following sections, we will study network security from several angles.

1. Security Concepts

"Security" is an all-encompassing term that describes all the concepts, techniques, and technologies to protect information from unauthorized access. There are several requirements for information security.

- Confidentiality. Hiding data, usually with encryption, to prevent unauthorized viewing and access.
- Authenticity. The ability to know that the person or system you are communicating with is who or what you think it is.
- Access control. Once a person or system has been authenticated, their ability to access data and use systems is determined by access controls.
- Data Integrity. Providing assurance that an information system or data is genuine.
- Availability. Making sure that information is available to users in a secure way.

2. Security Threats and Vulnerabilities

Information security threats are everywhere. Your internal users may be stealing data or inadvertently revealing sensitive passwords or other material to people who are attempting to infiltrate your organization. Attackers, usually refer as hackers, from the outside may be gaining access through dial-up Internet connections or external server-to-server connections.

An attack is an attempt to take control of a system (a computer, network server, Web site, and so on) using a variety of methods with the intent to take over the system, or simply to shut it down or prevent other people from accessing it (a denial-of-service attack). Attacks may also take place on cryptographic systems information that has been encrypted, such as password files, secure data files, and so on. There are two primary types of attacks:

- Passive attack. Monitoring and collecting information about a system to be used in a later attack, or to attack another related system.
- Active attack. An active attack is one in which the attacker actually attempts to gain access to a system through unauthorized or illegal means.

Hackers may monitor the sessions of other users (passive attack) and attempt to take over the session (active attack). In a replay attack, the attacker uses previously gathered information to gain access to a system by "replaying" it to the system, which thinks that it is dealing with a valid session.

3. Security Mechanisms

Trust is an important aspect of security. There are many different forms and levels of trust

between people and computer systems. Many exchanges and transactions on the Internet take place between people who have never met. A third party can provide this trust by verifying the authenticity of parties in an exchange. Traditionally, banks and escrow companies have provided this trust. On the Internet, it is provided by CAs (certificate authorities).

Trust management systems provide security services for users and free up applications from having to provide their own mechanisms for interpreting credentials, authentication, and policy. A trust management system can be queried by an application with questions about how trust should be handled.

Security policies are an essential part of an organization's general operations. The often-quoted rule of "denying what is not specifically permitted" provides a good basis for defining any security policy. While this rule usually applies to firewalls, it also provides a good approach to overall security.

(1) Physical Security Management

While security is usually associated with some form of cryptography, physical systems must be protected from theft, damage, and corruption. Data must be backed up. In addition, the availability of data is important. Systems must be kept online, even in the event of fires, floods, and earthquakes. Therefore, some means of replicating systems to other sites is often necessary.

(2) Cryptography, Keys, and Certificates

There are a number of security mechanisms, most of which are based on some form of cryptography. These mechanisms allow secure data exchange over corporate networks and the Internet. They can be used to hide data, ensure the integrity of messages, and authenticate users or systems.

Cryptography provides the basis for securing data. An encryption algorithm is a mathematical routine that scrambles data, based on a user key, in a way that can be recovered with the same key or key pair. There are two types of encryption algorithms. There are symmetric secret-key algorithms and asymmetric public-key algorithms.

The advantage of the public-key scheme is that it eliminates the problems of key exchange. A trusted third party holds the public key and makes it available to other people in the form of a certificate. Certificate authorities bind a person's public key with validated information about that person, thus creating a digital certificate.

Certificates (and their keys) can be used to digitally sign messages. A signed message provides proof that a message is authentic, that it has not been tampered with, and that it has no errors.

The public-key cryptography scheme is an essential part of doing business on the Internet. By putting public keys in certificates, it is possible for parties who don't know each other to establish secure trusted connections. If both parties trust the certificates issued by a particular CA, then they trust the contents of those certificates. The public keys can then be used for authentication and to establish encrypted communication sessions. A PKI (public-key infrastructure) is an organized hierarchical structure (potentially global) for creating, managing, and distributing certificates.

(3) Securing Connections

A number of protocols exist to secure the connection between systems. Some of these protocols also provide authentication features. For example, PPP (Point-to-Point Protocol) includes the ECP (Encryption Control Protocol), which provides a method to negotiate an encryption method between the two points. Secure connections across the Internet can be implemented with VPN technology. IPSec (IP Security) has emerged as the most important protocol for establishing secure connections.

4. Security-Related Organizations

There are a variety of security specifications and initiatives, some developed by vendors and some developed by consortiums. Governments also define security specifications. Table 14-1 lists some security-related organizations and their websites.

Table 14-1 Security-Related Organizations

NIST (National Institute of Standards and Technology)	http://www.nist.gov/
NSA (National Security Agency)	http://www.nsa.gov/
CERT (Computer Emergency Response Team)	http://www.cert.org/
CIAC (Computer Incident Advisory Capability)	http://ciac.llnl.gov/
FIRST (Forum of Incident Response and Security Teams)	http://www.first.org/
EFF (Electronic Frontiers Foundation)	http://www.eff.org/
NetSec Int'l (Network Security International Association)	http://www.netsec-intl.com/

14.2 Information Security Technology

14.2.1 Information Authentication Technology

1. Authentication Protocols

Authentication is the technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter. Verifying the identity of a remote process in the face of a malicious, active intruder is surprisingly difficult and requires complex protocols based on cryptography. In this section, we will study some of the many authentication protocols that are used on insecure computer networks.

As an aside, some people confuse authorization with authentication. Authentication deals with the question of whether or not you are actually communicating with a specific process. Authorization is concerned with what that process is permitted to do. For example, a client process contacts a file server and says, "I am Scott's process and I want to delete the file cookbook.old." From the file server's point of view, two questions must be answered:

- Is this actually Scott's process (authentication)?
- Is Scott allowed to delete the file cookbook.old (authorization)?

Only after both questions have been unambiguously answered in the affirmative can the requested action take place. The former question is really the key one. Once the file server knows whom it is talking to, checking authorization is just a matter of looking up entries in local tables.

2. Digital Signatures

The authenticity of many legal, financial, and other documents is determined by the presence or absence of an authorized handwritten signature. And photocopies do not count. For computerized message systems to replace the physical transport of paper and ink documents, a solution must be found to these problems.

One approach to digital signatures called secret-key signatures is to have a central authority that knows everything and whom everyone trusts, says Big Brother (BB). Each user then chooses a secret key and carries it by hand to BB's office. Thus only Alice and BB know Alice's secret, K_A, and so on.

A structural problem with using secret-key cryptography for digital signatures is that everyone has to agree to trust Big Brother. Furthermore, Big Brother gets to read all signed messages. The most logical candidates for running the Big Brother server are the governments, the banks, or the lawyers. These organizations do not inspire total confidence in all citizens. Hence it would be nice if signing documents did not require a trusted authority. Fortunately, public-key cryptography can make an important contribution here. It is public-key signatures.

There are also some social issues, such as the implication of network security for individual privacy and society in general, patents. And network security is politicized to an extent few other technical issues are, and rightly so, since it relates to the difference between a democracy and a police state in the digital era.

14.2.2 Encryption Technology

1. Traditional Cryptography

Until the advent of computers, one of the main constraints on cryptography had been the ability of the code clerk to perform the necessary transformations, often on a battlefield with little equipment. An additional constraint has been the difficulty in switching over quickly from one cryptographic method to another one, since this entails retraining a large number of people. However, the danger of a code clerk being captured by the enemy has made it essential to be able to change the cryptographic method instantly, if need be.

The messages to be encrypted, known as the plaintext, are transformed by a function that is parametrized by a key. The output of the encryption process, known as the ciphertext, is then transmitted, often by messenger or radio. We assume that the enemy, or intruder, hears and accurately copies down the complete ciphertext. However, unlike the intended recipient, he does not know what the decryption key is and so cannot decrypt the ciphertext easily. Sometimes the intruder can not only listen to the communication channel (passive intruder) but can also record

messages and play them back later, inject his own messages, or modify legitimate messages before they get to the receiver (active intruder). The art of breaking ciphers is called cryptanalysis. The art of devising ciphers (cryptography) and breaking then (cryptanalysis) is collectively known as cryptology.

The real secrecy is in the key, and its length is a major design issue. Consider a simple combination lock. The general principle is that you enter digits in sequence. Everyone knows this, but the key is secret. A key length is two digits that means there are 100 possibilities. A key length of three digits means 1000 possibilities, and a key length of six digits means a million. The longer the key, the higher the work factor the cryptanalyst has to deal with. The work factor for breaking the system by exhaustive search of the key space is exponential in the key length. Secrecy comes from having a strong (but public) algorithm and a long key. To prevent your kid brother from reading your E-mail, 64-bit keys will do. To keep major governments at bay, keys of at least 256 bits are needed.

2. Two Fundamental Cryptographic Principles

Although we will study many different cryptographic systems in the pages ahead, there are two principles underlying all of them that are important to understand.

The first principle is that all encrypted messages must contain some redundancy, that is, information not needed to understand the message. An exam may make it clear why this is needed. Consider a mail-order company, The Couch Potato (TCP), with 60000 products. Thinking they are being very efficient, TCP's programmers decide that ordering messages should consist of a 16-byte customer name followed by a 3-byte data field (l byte for the quantity and 2 bytes for the product number). The last 3 bytes are to be encrypted using a very long key known only by the customer and TCP.

At first this might seem secure, and in a sense it is because passive intruders cannot decrypt the messages. Unfortunately, it also has a fatal flaw that renders it useless. Suppose that a recently fired employee wants to punish TCP for firing her. Just before leaving, she takes (part of) the customer list with her. She works through the night writing a program to generate fictitious orders using real customer names. Since she does not have the list of keys, she just puts random numbers in the last 3 bytes, and sends hundreds of orders off to TCP.

When these messages arrive, TCP's computer uses the customer's name to locate the key and decrypt the message. Unfortunately for TCP, almost every 3-byte message is valid, so the computer begins printing out shipping instructions. While it might seem odd for a customer to order 137 sets of children's swings, or 240 sandboxes, for all the computer knows, the customer might be planning to open a chain of franchised playgrounds. In this way an active intruder (the ex-employee) can cause a massive amount of trouble, even though she cannot understand the messages her computer is generating.

This problem can be solved by adding redundancy to all messages. However, adding redundancy also makes it much easier for cryptanalysts to break messages.

Thus cryptographic principle number one is that all messages must contain redundancy to prevent active intruders from tricking the receiver into acting on a false message. However, this same redundancy makes it much easier for passive intruders to break the system, so there is some tension here. Furthermore, the redundancy should never be in the form of *n* zeros at the start or end of a message, since running such messages through some cryptographic algorithms gives more predictable results, making the cryptanalysts' job easier. A random string of English words would be a much better choice for the redundancy.

The second cryptographic principle is that some measures must be taken to prevent active intruders from playing back old messages. If no such measures were taken, our ex-employee could tap TCP's phone line and just keep repeating previously sent valid messages.

3. Secret-Key Algorithms

Modern cryptography uses the same basic ideas as traditional cryptography, transposition and substitution, but its emphasis is different. Traditionally, cryptographers have used simple algorithms and relied on very long keys for their security. Nowadays the reverse is true: the object is to make the encryption algorithm so complex and involuted that even if the cryptanalyst acquires vast mounds of enciphered text of his own choosing, he will not be able to make any sense of it at all. Transpositions and substitutions can be implemented with simple circuits.

- DES. In January 1977, the U.S. government adopted a product cipher developed by IBM as its official standard for unclassified information. This cipher, DES (Data Encryption Standard), was widely adopted by the industry for use in security products. It is no longer secure in its original form, but in a modified form it is still useful.
- IDEA. IDEA (International Data Encryption Algorithm) was designed by two researchers in Switzerland. It uses a 128-bit key, which will make it immune to brute force, and also to withstand differential cryptanalysis. No currently known technique or machine is thought to be able to break IDEA.

4. Public-Key Algorithms

Historically the key distribution problem has always been the weak link in most cryptosystems. No matter how strong a cryptosystem was, if an intruder could steal the key, the system was worthless. Since all cryptologists always took for granted that the encryption key and decryption key were the same (or easily derived from one another) and the key had to be distributed to all users of the system, it seemed as if there was an inherent built-in problem: keys had to protected from theft, but they also had to be distributed so they could not just be locked up in a bank vault.

Public-key cryptography requires each user to have two keys: a public key, used by the entire world for encrypting messages to be sent to that user, and a private key, which the user needs for decrypting messages. We will consistently refer to these keys as the public and private keys, respectively, and distinguish them from the secret keys used for both encryption and decryption in conventional (also called symmetric key) cryptography.

Although the RSA algorithm known by the initials of the three discovers (Rivest, Shamir, Ademan) is widely used, which method is based on some principles from number theory, it is by no means the only public-key algorithm known. The first public-key algorithm was the knapsack algorithm (Merkle and Hellman, 1978). Other public-key schemes are based on the difficulty of computing discrete logarithms (Rabin, 1979). Algorithms that use this principle have been invented by El Gamal (1985) and Schnorr (1991). A few other schemes exist, such as those based on elliptic curves (Menezes and Vanstone, 1993), but the three major categories are those based on the difficulty of factoring large numbers, computing discrete logarithms, and determining the contents of a knapsack from its weight. These problems are thought to be genuinely difficult to solve because mathematicians have been working on them for many years without any great breakthroughs.

14.2.3 Database Security

Selectively protecting the most sensitive data that is at rest in databases from unauthorized access is critical, since that is where 90 percent of sensitive information resides. There is an important distinction between network security and data security. Database security does not supersede other security technologies, such as network-layer firewalls, network monitoring, SSL-secured communications, operating system and application hardening. But data protection needs to be in place as the core element of a complete enterprise security infrastructure. However, often companies do not realize the potential amount of risk associated with sensitive information within databases until they run an internal audit which details who has access to sensitive data.

The 2002 Computer Security Institute (CSI) Computer Crime and Security Survey revealed that over half of the databases have some kind of breach on a yearly basis and the average breach is close to $4 million in losses. You want to wrap each individual data item in a protective security, rather than simply building a firewall fence around the database. One of the best ways to develop an effective database security is recognizing that securing data is essential to a company's reputation, profitability and critical business objectives.

The database security solution also protects information on back-up tapes that are stored off-site. When considering ways to protect sensitive database information, it's important to ensure that the privacy protection process does not prevent authorized persons from obtaining the right data at the appropriate times. It is important that your database security solution is application transparent.

You can assign all sorts of rights, logins, roles and passwords to restrict queries and application usage. IT security experts often recommend selectively encrypting and securing sensitive database information at the data-item level to ensure excellent performance.

Basically database security can be broken down into the following key points of interest.

- Server Security
- Database Connections
- Table Access Control
- Restricting Database Access

14.3 Computer Virus

Computer viruses are self-replicating (copies itself) programs deliberately designed to interfere with computer operation, record, corrupt, or delete data, or spread themselves to other computers and throughout the Internet.

When the code of a virus is executed, it spreads itself to other programs. A typical computer virus does two things. First, it copies itself into previously uninfected programs. Second, it executes other instructions that the virus creator has included in it. Some viruses don't have any harmful instructions at all. Instead, they cause damage by replicating and taking up disk space.

Many people believe that any program that does something malicious to a computing system is a virus. This isn't true unless that program can reproduce itself. Computer viruses can cause the loss or change of programs or data. Viruses can spread from one program to another and from system to system.

Viruses often slow things down and cause other problems in the process.

14.3.1 Virus Behavior

A virus contains a "self-replicating code" which enables it to reproduce itself. The code is a program that copies itself to other programs. When this code is executed the virus "infects" another program by attaching a copy of its code to that program. Then when that program is executed the viral code is executed as well and its copies are placed in other programs, thus the virus spreads throughout the system.

Some viruses infect programs when they are executed. Some viruses, however, only infect when there is a certain trigger. A trigger is a condition that prompts a virus to perform a certain action. Triggers can be a time, day, a certain keystroke on the keyboard, or anything.

Aside from infecting a system, viruses usually also attack the system. A virus' attack phase is also triggered by some event. The attack can cause serious damage or be a nuisance, depending on what the virus' creator has instructed it to do. Some viruses can delete files or change data while other less harmful viruses play music or display messages on the screen. Not all viruses contain instructions to attack. Some just reproduce and never go beyond that. However, as they replicate, they take up storage and can possibly damage programs or disks.

Viruses usually try to remain undetected by the user and antivirus software in order to have time to spread throughout a system. Some viruses will only attack after they have infected most of the system. This way they can avoid detection until they've become sufficiently planted within the system. A virus can also mark the files that it has already infected. By doing this the virus avoids reinfecting a file, if it did, the file would grow larger. A file that changes size would attract the attention of the user or antivirus software. Viruses will use many techniques to avoid being detected.

14.3.2 Types of Viruses

There are two major categories of viruses.

1. Boot Sector Viruses

Boot sector viruses infect the boot sector of a disk, therefore affecting the computer system during the start-up process. The boot sector is a small area on a disk which is read by the computer when it is turned on (booted). The most common way for a system to become infected is when an infected disk is left in the disk drive when the computer is being booted. The computer will read the virus from the boot sector and infect the entire system. These viruses are difficult to deal with because they are read during the start-up process even before the system is able to load its antivirus software to detect the virus.

2. File Infector Viruses

File infector viruses affect the program files that a system must load in order to make software function. The virus infects a file (usually files ending in ".COM" or ".EXE") by overwriting part of the program. By overwriting the program the virus replaces part of the program code with viral code. When the program is executed, the virus code is executed and it infects more files.

Here are some other types of viruses: Memory Resident Viruses, Non-memory Resident Viruses, Multi-Partite Viruses, Stealth Viruses, and Polymorphic Viruses.

14.3.3 Virus Prevention and Detection

Although there isn't a foolproof way to prevent a virus infection, there are ways to reduce the risk of an infection.

1. User Education

Users should be aware of the dangers of viruses. They need to know how viruses are spread in order to ensure that they don't do anything that places their computers in danger of becoming infected. Users should know not to share programs or disks and to not boot a system with an infected disk. They should know enough about viruses to be able to protect themselves from common viruses that can easily be avoided with some general knowledge.

Users should take note of any odd behavior in a computer system. Odd behavior may be an indication of an infection. However, it's also important to realize that any strange behavior can also be caused by bugs, user errors and hardware problems. Therefore common sense and good judgment must also be used.

Some observable symptoms of viruses are:

- programs taking longer to start or running slower than usual.
- sudden changes in file size.
- an unexplained decrease in the amount of memory.
- Unusual things appearing on the screen.
- files suddenly disappearing.

2. Isolated Systems

Viruses can spread through systems when information is shared. Systems that are connected

to other systems by a network can spread viruses across the network. A way to prevent this from happening is to isolate the system. An isolated system can't become infected or infect other systems if they aren't connected.

3. Checking and Antivirus Software

Any and all software should be checked for viruses before being used. Software that is used by many people should be regularly checked for infection. Even newly purchased software should be tested in case the distributor unwittingly distributes infected products. Antivirus softwares are also very useful in preventing viruses.

14.4 Firewall

14.4.1 Concept of Firewall

If you spend a lot of time on the Internet and you are not behind a firewall, then you are living on borrowed time. Putting some protection between you and the Internet is probably the third most important thing that you can do (after getting virus checking software and performing regular backups).

A firewall is a security system designed to protect an organization's network against external threats. It consists of hardware and software that control access to a company's intranet or other internal networks.

What a personal firewall does is isolate your computer from the rest of the Internet. It does this by inspecting each packet of data to determine if it should be allowed to get to (and in some cases from your machine). The best protection completely hides your computer — this is called stealth mode.

Typically organizational firewalls include a special computer called a proxy server. This computer is a gatekeeper. All communications between the company's internal networks and the outside world must pass through it. By evaluating the source and the content of each communication, the proxy server decides whether it is safe to let a particular message or file pass into or out of the organization's network.

The proxy server can be viewed as a gateway between two networks, usually a private internal network and the Internet. It is sometimes called an application-level gateway. The proxy server hides the internal network from the external network. It keeps hackers from accessing or even knowing about internal IP addresses.

A proxy server runs as software on a computer and acts on behalf of a client to make requests outside the client's network. For example, when an internal user attempts to access the Internet, the proxy server intercepts the request and makes the request itself. The internal user never makes a direct request to an outside system. Likewise, when the Internet server returns a response, it is intercepted by the proxy server and transferred to the user. The proxy server can filter all incoming packets and discard any that are not related to an internal request. This prevents hackers from attacking internal systems.

Proxy servers are both firewall and caching systems. Since they provide a centralized location where internal users access the Internet, the proxy server can cache frequently accessed documents from sites on the Internet and make them quickly available for other internal users that need the documents.

14.4.2 Types of Firewall

You have the option of installing a software firewall or a hardware firewall.

1. Software Firewall

A software firewall runs on your computer system in the background. It intercepts each network request and determines if the request is valid or not. Software firewalls offer the following advantages:

- They are generally very inexpensive.
- They are very easy to configure.

They have the following disadvantages:

- Since they run on your computer they require resources (CPU, memory and disk space) from your system.
- They can introduce incompatibilities into your operating system.
- You must install exactly the correct version for your operating system.
- You must purchase one copy for each system on your home network.

2. Hardware Firewall

A hardware firewall is generally a small box which sits between your computer and your modem. In general, hardware firewalls have the following advantages:

- They tend to provide more complete protection than software firewalls.
- A hardware firewall can protect more than one system at a time.
- They do not affect system performance since they do not run on your system.
- They are independent of your operating system and applications.

They have the following disadvantages:

- They tend to be expensive, although if you have a number of machines to protect it can cost less to purchase one hardware firewall than a number of copies of a software product.
- Since they do not run on your computer, they can be challenging to configure.

3. Firewall mixture

The best protection is a combination of both hardware and software firewalls. This is the ideal, since both have different advantages and disadvantages.

14.4.3 Firewall Implementation

In general, there are three types of firewall implementations, some of which can be used together to create a more secure environment. These implementations are: packet filtering, application proxies, and circuit-level or generic-application proxies.

Packet filtering is often achieved in the router itself. Application proxies, on the other hand, usually run on standalone servers. Proxy services take a different approach than packet filters, using a (possibly) modified client program that connects to a special intermediate host that actually connects to the desired service.

1. Packet Filtering

Consider your network data a neat little package that you have to deliver somewhere. This data could be part of an E-mail, file transfer, etc. With packet filtering, you have access to deliver the package yourself. The packet filter acts like a traffic cop; it analyzes where you are going and what you are bringing with you. However, the packet filter does not open the data package, and you still get to drive it to the destination allowed.

Most commercial routers have some kind of built-in packet filtering capability. However some routers that are controlled by ISPs may not offer administrators the ability to control the configuration of router. In those cases, administrators may opt to use a standalone packet filter behind the router.

Either way, an administrator needs to understand how to identify data packages in terms the packet filter can understand. Since all Internet traffic is based on IP (Internet Protocol), each application or “package” can be identified through a specific TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) port. These ports are registered and defined in RFC (Request for Comment) 1700 which can be found on the Internet. For example, port 23 is for Telnet. A company could block incoming packets for all IP addresses combined with port 23. In this way, no one outside the company could log in via Telnet.

2. Application Proxy

To understand the application proxy, consider this scenario where you need to deliver your neat little package of network data. With application-level proxies, the scenario is similar, but now you need to rely on someone else to deliver the package for you. Hence the term proxy illustrates new scenario. The same rules apply as they do for packet filtering, except that you don’t get to deliver your package past the gate. Someone will do it for you, but that agent needs to look inside the package first to conform its contents. If the agent has permission to deliver the contents of the package for you, he will.

Most commercial routers do not have proxy capabilities today, although we believe that proxy technology will be integrated with router code in the future. Until then, you need to rely on a standalone system that can support application-level proxy services.

Since an application proxy needs to communicate on behalf of the sender, it needs to understand the specific language or protocols associated with a particular application. Take as an example the widely used HTTP (Hyper Text Transfer Protocol) proxy. If you are using a browser on your network, it is highly likely that your IS group has an HTTP proxy configured to allow you to access the Web via a central server. That single machine understands HTTP conversations and

can speak on behalf of the requesting client. This is application-level proxy.

Of course, security and encryption also come into play, since the proxy must be able to open the "package" to look at or decode its contents. These are important issues obviously, but to do them justice would require another article.

3. Circuit-Level or Generic-Application Proxy

As with application-level proxies, you need to rely on someone to deliver your package for you. The difference is that if these circuit-level proxies have access to deliver the package to your requested destination, they will. They do not need to know what is inside.

Circuit-level proxies (specifically SOCKS) work outside of the application layers of the protocol. These servers allow clients to pass through this centralized service and connect to source address of connection requests and can block unauthorized clients from connecting out onto recompiling and linking them with a SOCKS client library. DLL-based TCP stacks have the use of shims, eliminating the need to recompile.

14.5 Standards of Information Security

1. IPSec

The standards were developed by the Internet Engineering Task Force's (IETF) to secure communications over both public and private networks, though it is particularly beneficial to public networks.

IPSec is a framework that is built into various security products to provide end-to-end security in wide area networking communications. Using strong encryption, and public key cryptography, IPSec can secure data links.

2. SET

Secure Electronic Transactions (SET) is an open protocol which has the potential to emerge as a dominant force in the securing of electronic transactions. Jointly developed by Visa and MasterCard, in conjunction with leading computer vendors such as IBM, SET is an open standard for protecting the privacy, and ensuring the authenticity, of electronic transactions. This is critical to the success of electronic commerce over the Internet.

3. SSE-CMM

The Systems Security Engineering Capability Maturity Model (SSE-CMM) was developed to advance security engineering as a defined, mature, and measurable discipline. It describes the characteristics essential to the success of an organization's security engineering process, and is applicable to all security engineering organizations including government, commercial, and academic. Its acceptance as ISO/IEC 21827, makes it the first formal standard of this scale

dedicated to security engineering practices covering.

There also are some other standards of information security, such as SSL/TLS, BS 7799, ISO 13335 and so on. And we will not introduce them here.

14.6 E-Commerce Security

14.6.1 Significance of Corporate Information Security

In today's Internet world, it is relatively easy to create, alter and transmit information. The advancement in computing capacity and interconnectivity has presented a situation where small efforts can cause potentially large losses. Both accidental and intentional breaches are easier and more likely. This is a major challenge to businesses that want to take advantage of the current information technology. Concern for information security is fairly widespread. According to Information Week Research's Global Information Security Survey conducted in June, 2000, nearly three-quarters of information security professionals regard security as a top priority, up from 56% two years ago. Those in banking, health care, finance, and telecommunications rate information security as the highest business priority, with retailers a little less concerned. In every sector, security is regarded as a key business driver.

A viable security policy should have the following characteristics:

- The policy must be clear and concise.
- The policy must have built-in incentives to motivate compliance.
- Compliance must be verifiable and enforceable.
- Systems must have good control for legitimate use: access, authentication, and authorization.
- There must be regular backup of all critical data.
- There must be a disaster recovery and business continuity plan.

14.6.2 Current Processes and Tools for Implementing E-Business Security

A typical three-tier e-business architecture comprises the client, web and commerce servers, and database servers. A systematic implementation of e-business security must ensure that each of these components is secure. This requires security policy and implementation at three levels: network security, system level security and transaction level security.

The current common e-business security practice translates into acquiring sophisticated servers, firewall software, intrusion detection systems, and obtaining digital certificates.

This is the area of security risk management that is principally a technology issue. Each component has to be addressed with a view to implementing a complete e-business secure infrastructure. Notable elements in that strategy will include cryptography, PKI and digital signature technology. This is where the system information security officer can go over a checklist of what is necessary and what the organization has.

A typical checklist will include:

physical protection for computers	antivirus software
network systems management	digital certificate
E-mail control security	strong authentication
networks security	access control
firewalls	audit and tracing software
encryption	backup and disaster recovery
PKI	biometric software
incident handling	wireless communications security

Technical Notes to the Text

1. information security，信息安全。信息安全指信息网络的硬件、软件及其系统中的数据受到保护，不受偶然的或者恶意的原因而被破坏、更改、泄露，系统连续、可靠、正常地运行，信息服务不被中断。

2. hacker，黑客，源于英语 hack，原意是“劈砍”。黑客是指一些计算机编程高手，能够发现系统的安全漏洞。

3. network security，网络安全。在分布式计算环境中，对信息的传输、存储、访问提供安全保护，以防止信息被窃取、篡改和非法操作。

4. secrecy，保密。保密就是保护信息不泄露或不暴露给那些未授权掌握这一信息的实体，如人或组织。

5. authentication，鉴别。由于网络电子商务交易系统的特殊性，企业或个人的交易通常都是在虚拟的网络环境中进行的，所以对个人或企业实体进行身份确认成了电子商务中很重要的一环。

6. CA (certificate authorities)，认证中心。认证中心是认证机构的国际通称，是指对数字证书的申请者发放、管理、取消数字证书的机构。CA 的作用是检查证书持有者身份的合法性，并签发证书(在证书上签字)，以防证书被伪造或篡改。

7. cryptography，密码学。密码学以研究数据保密为目的，对存储或传输的信息采取秘密交换以防止第三者对信息的窃取。

8. PKI (public key infrastructure)，公钥基础设施。它是一种遵循既定标准的密钥管理平台，能够为所有网络应用提供加密和数字签名等密码服务及所必需的密钥和证书管理体系。

9. digital signatures，数字签名。数字签名的主要目的是识别信息来源，本身不具备对信息加密的功能。

10. plaintext，明文。明文指需要加密的报文。

11. key，密钥。用于加密和解密的钥匙，称为密钥。

12. ciphertext，密文。密文指加密以后形成的报文。

13. cryptanalysis，密码分析。在不知道密码系统密钥的情形下，将密文转变成明文的步骤和操作(过程)。

14. cryptographic principle，加密原则。加密就是把数据信息(即明文)转换为不可辨识的

形式(即密文)的过程。目的是使不应了解该数据信息的人无法知道和识别。

15. DES (data encryption standard)，数据加密标准。这种算法被美国国家标准局于 1977 年 1 月 5 日正式确定为美国的统一数据加密标准。

16. RSA algorithm，RSA 算法。RSA 算法是基于大素数分解的困难性而开发的算法。

17. symmetric key，对称密钥。传统加密方法的共同特点是采用单钥技术，即加密和解密过程中使用同一密钥，所以它也被称为对称式加密方法；该密钥也被称为对称密钥。

18. public-key signature，采用公开密钥的数字签名。目前，通用的办法是采用建立在 PKI(公钥基础设施)基础之上的数字证书，通过把要传输的数字信息进行加密和签名，保证信息传输的机密性、真实性、完整性和不可否认性，从而保证信息的安全传输。

19. computer virus，计算机病毒。它是一种能够自我复制的程序，具有干扰计算机运行、记录、修改或者删除数据，并且能够向其他计算机和互联网传播的能力。

20. firewall，防火墙。防火墙是加强 Internet 与 Intranet(内部网)之间安全防范的一个或一组系统。

21. packet filtering，包过滤。包过滤型防火墙可以动态检查流过的 TCP/IP 报文头，检查报文头中的报文类型、源 IP 地址、目的 IP 地址、源端口号等域，根据事先定义的规则，决定哪些报文允许通过，哪些报文禁止通过。

22. proxy server，代理服务器。代理服务器即提供网络代理服务的服务器。

Word Bank to the Text

A. Useful new words

loom	*v*. 隐现，迫近
methodology	*n*. 方法学，方法论
genuine	*adj*. 真实的，真正的
inadvertent	*adj*. 疏忽的，漫不经心的
infiltrate	*v*. 渗透
escrow	*n*. 契约，合同
credential	*n*. 信任状，证书
corruption	*n*. 腐败，贪污
authenticate	*v*. 鉴别
scramble	*v*. 搅乱，使混杂
eliminate	*v*. 排除，消除
encrypt	*v*. 加密，将……译成密码
confidentiality	*n*. 机密性
decompose	*v*. 分解
disclosure	*n*. 透露，显露
disruption	*n*. 中断，分裂
imposter	*n*. 造假者

malicious	*adj*. 怀恶意的
intruder	*n*. 入侵者
cryptography	*n*. 密码术
aside	*n*. 旁白
unambiguous	*adj*. 不含糊的，明确的
entail	*v*. 需要
ciphertext	*n*. 密码，暗记文
exponential	*adj*. 指数的，幂数的
fictitious	*adj*. 假想的，编造的
bankruptcy	*n*. 破产
franchise	*v*. 赋予特权
redundancy	*n*. 冗余
involuted	*adj*. 纷乱的，复杂的
withstand	*v*. 抵挡，经受住
deliberately	*adv*. 故意地
foolproof	*adj*. 十分安全的，极坚固的
unwitting	*adj*. 无意的，无心的
intercept	*v*. 中途阻止，截取
vault	*n*. 保险库
discrete	*adj*. 不连续的，离散的
supersede	*v*. 代替，取代
breach	*n*. 违背，破坏
profitability	*n*. 收益性，利益率
transparent	*adj*. 透明的，显然的
drawbridge	*n*. 吊桥
incentive	*n*. 刺激，鼓励
compliance	*n*. 依从，顺从

B. Useful expressions

derive from	得自，由来
lie in	在于
regardless of	不管，不顾
with the intent to	试图做某事
take over	接收，接管
free from	不受……拘束的
tamper with	损害，篡改
fall into the hands of	落入……的控制
confuse with	混淆
in sequence	顺次，挨次

switch over	转换
keep at bay	阻止，挡住
in a sense	在某种意义上
trick into	哄骗
be immune to	对……有免疫的，不受影响的
take it for granted	视为当然
associate with	联合
on a yearly basis	每年一次地
break down into	分类
aside from	除……以外
attract the attention of	吸引……注意力
on behalf of	代表
insert into	插入
integrate with	使与……结合
take as an example	以……为例
come into play	开始活动
block from	阻止
in conjunction with	和……一道，结合
be applicable to	适用于
dedicate to	献身于
take advantage of	利用

C. Technical terms and proper names

network security	网络安全
confidentiality	机密性
authenticity	真实性
access control	访问控制
data integrity	数据完整性
availability	可用性
active attack	主动攻击
passive attack	被动攻击
secrecy	保密
authentication	鉴别
plaintext	明文
key	密钥
ciphertext	密文
cryptanalysis	密码分析
cryptology	密码学
intruder	入侵者

work factor	工作因子
algorithm	算法
redundancy	冗余
cryptographic principle	加密原则
DES (data encryption standard)	数据加密标准
IDEA (international data encryption algorithm)	国际数据加密算法
secret-key algorithm	私有密钥算法
public-key algorithm	公开密钥算法
RSA algorithm	RSA 算法
symmetric key	对称密钥
asymmetric public-key	非对称公开密钥
knapsack algorithm	背包算法
discrete logarithm	离散对数
authentication protocol	鉴别协议
authorization	授权
digital signature	数字签名
secret-key signature	采用秘密密钥的数字签名
public-key signature	采用公开密钥的数字签名
individual privacy	个人隐私
virus	病毒
replicate	复制
firewall	防火墙
packet filtering	包过滤
proxy	代理
application proxy	应用程序级代理
generic-application proxy	通用应用程序代理
circuit-level	电路级
RFC (request for comments)	请求评注
DLL (dynamic link library)	动态链接库
e-commerce	电子商务
intrusion detection	入侵检测

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. Once a person or system has been authenticated, their ability to access data and use systems

is determined by ________.

2. All encrypted messages must contain some ________.

3. Public-key cryptography requires each user to have two keys: a ________ key, and a ________ key.

4. There are several requirements for information security: confidentiality, ________, access control, ________, availability.

5. ________ services take a different approach than packet filters, using a (possibly) modified client program that connects to a special intermediate host that actually connects to the desired service.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. Active attack means monitoring and collecting information about a system to be used in a later attack, or to attack another related system. (　)

2. Public-key cryptography requires each user to have a key: a public key. (　)

3. Authentication deals with the question that you are actually communicating with a specific process. (　)

4. One approach to digital signatures called secret-key signatures is to have a central authority that knows everything and whom everyone trusts, says Big Brother. (　)

5. Modern cryptography uses the same basic ideas as traditional cryptography, transposition and substitution. (　)

Ⅲ. Match each of the following terms with the appropriate definition.

packet filter	computer virus	authentication
SET	firewall	

1. ________ Which is the technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter.

2. ________ Which is a security system designed to protect an organization's network against external threats.

3. ________ Which is a program designed to replicate and spread on its own, generally with the victim being oblivious to its existence.

4. ________ To acts like a traffic cop and analyzes where you are going and what you are bringing with you.

5. ________ An open standard for protecting the privacy, and ensuring the authenticity, of electronic transactions.

Ⅳ. Answer the following questions.

1. List some information security technology.

2. What's the computer virus?

3. List two main types of computer viruses.
4. List three types of firewall implementations.

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

entail	discrete	profitability
Deliberate	incentive	fictitious
withstand	redundancy	loom

1. The threat of renewed civil war ______ ahead in the small island country.
2. Producing a series TV play ______ a lot of work.
3. Many journalists are interested in the source of these ______ rumours.
4. Exercise really can help you ______ stresses and strains more easily.
5. Two hundred ______ were announced in the shipyards.
6. The energy of the particle can only have certain special ______ values.
7. It has a ______ policy to introduce world art to Britain.
8. Many companies in Britain are keen on the idea of tax ______ for R & D.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "amass" and fill in the sentences with the right word.

Example: Governments, military, financial institutions, hospitals, and private businesses **amass** a great deal of confidential **information** about their employees, customers, products, research, and financial status.

1. They amassed enough ev______ to convict him on six charges.
2. It was better not to enquire too closely into how he had amassed his fo______.
3. During each war the monopoly capitalists amassed fabulous we______.
4. Case workers, as well as group workers, amass many details of kn______ about the customs of the individuals in whom they are interested.
5. It would be possible to amass many biblical re______ in support of this thesis.
6. A detective must find it as important as a novelist to amass his trivial ma______ before picking out the right clue.

Summary of the Text

Ⅶ. Choose the best one of the four Choices given to fill in each blank.

In today's Internet world, it is relatively easy to create, alter and transmit information. The advancement in computing capacity and interconnectivity has presented a situation where small efforts can cause __1__ large losses. Both accidental and __2__ breaches are easier and more likely.

This is a major challenge to businesses that want to take __3__ of the current information technology. Concern __4__ information security is fairly widespread.

The field of information security has grown and __5__ significantly in recent years. Information security means protecting information and information systems from __6__ access, use, disclosure, disruption, modification, or destruction. The terms information security, computer security and information assurance are frequently used __7__. These fields are interrelated and share the __8__ goals of protecting the confidentiality, integrity and availability of information.

Protecting confidential information is a business __9__, and in many cases also an ethical and legal requirement. For the individual, information security has a significant __10__ on privacy, which is viewed very differently in different cultures.

1. A. potentially	B. potential	C. more	D. much
2. A. intent	B. intention	C. intending	D. intentional
3. A. disadvantage	B. use	C. usage	D. advantage
4. A. for	B. about	C. in	D. on
5. A. evolve	B. evolved	C. evolving	D. to evolve
6. A. authorized	B. illegal	C. unauthorized	D. legal
7. A. changeable	B. changeably	C. interchangeably	D. interchangeable
8. A. common	B. same	C. regular	D. ordinary
9. A. requirement	B. request	C. standard	D. regulation
10. A. affect	B. effect	C. affection	D. effects

Translation

Ⅷ. Translate the following into Chinese.

1. Information security is concerned with the confidentiality, integrity and availability of data regardless of the form the data may take: electronic, print, or other forms.

2. Authentication is the technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter.

3. Thus cryptographic principle number one is that all messages must contain redundancy to prevent active intruders from tricking the receiver into acting on a false message.

4. When considering ways to protect sensitive database information, it's important to ensure that the privacy protection process does not prevent authorized persons from obtaining the right data at the appropriate times.

5. Both accidental and intentional breaches are easier and more likely. This is a major challenge to businesses that want to take advantage of the current information technology.

Chapter 15
Information System

Pre-reading Questions

1. Identify and describe the three levels of the organizational hierarchy. Which types of information systems serve each level?

2. Compare and contrast the MIS with the DSS.

3. Describe the functions, structure and characteristics of MIS and DSS.

15.1 Major Types of Information System in Organization

An information system can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization. In addition to supporting decision making, coordination, and control, information systems may also help managers and workers analyze problems, visualize complex subjects, and create new products.

Because there are different interests, and levels in an organization, there are different kinds of systems. No single system can provide all the information an organization needs. Figure 15-1 illustrates one way to depict the kinds of system found in an organization. In the illustration, the organization is divided into strategic, management, and operational levels and then is further divided into functional areas such as sales and marketing, manufacturing and production, finance and accounting, and human resources. Systems are built to serve these different organizational interests.

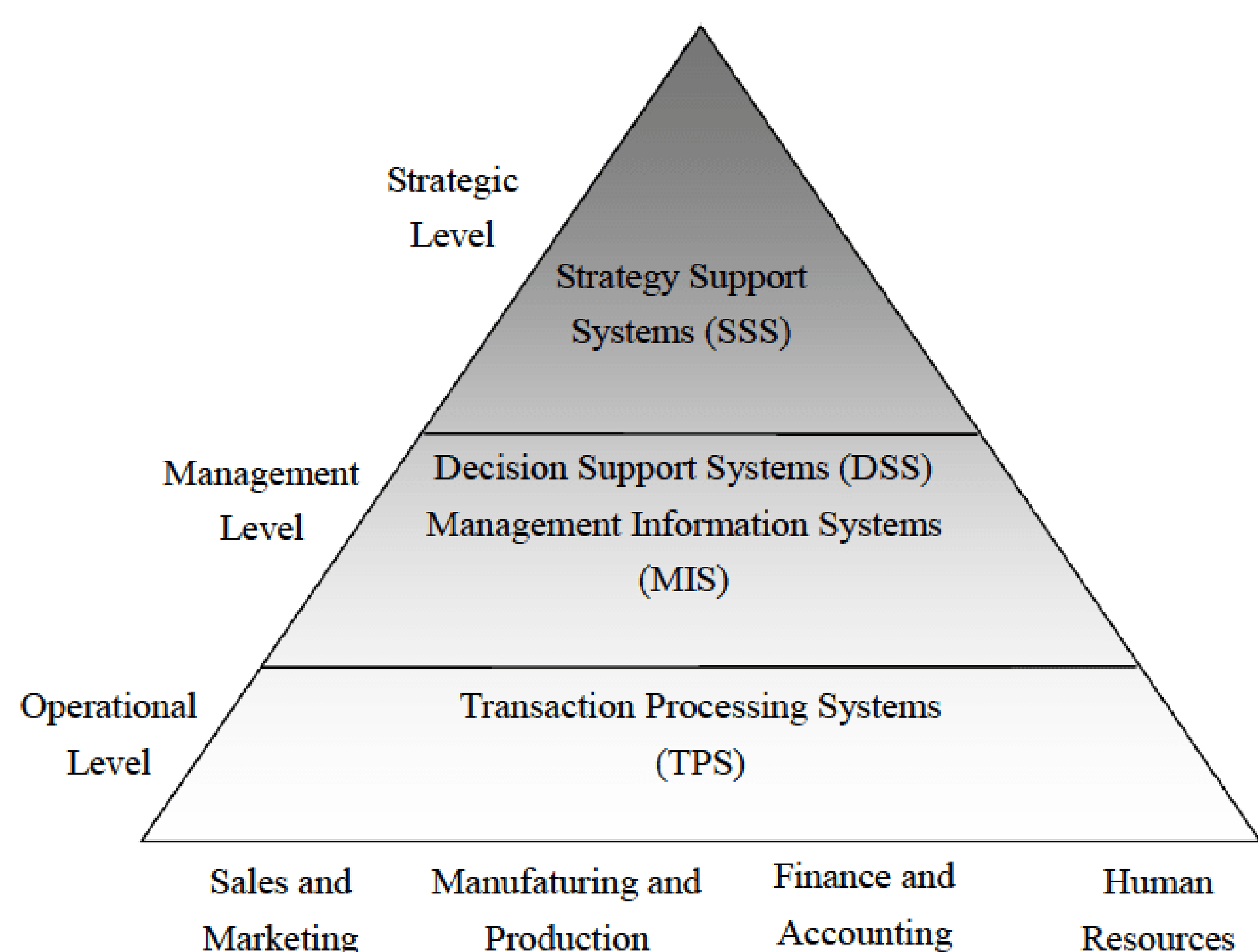


Figure 15-1 Types of Information Systems in Orgnizations

Figure 15-1 shows the specific types of information systems that correspond to each organizational level. The organization has strategy support systems (ESS) at strategic level; management information systems (MIS) and decision support systems (DSS) at the management level; and transaction

processing systems (TPS) at the operational level. Systems at each level in turn specialized to serve each of the major functional areas. Thus, the typical systems found in organizations are designed to assist workers or managers at each level and in the functions of sales and marketing, manufacturing and production, finance and accounting, and human resources.

15.2 Transaction Processing System (TPS)

There are various views of a transaction processing system (TPS). For example, there is the user's view, the programmer's view, the implementer's view, and the operator's view.

For a user, a TPS is usually viewed through an online transaction application program (see Figure 15-2). Hence, the user's view is often a set of menus and forms. The user selects a transaction from a transaction selection menu and then enters data in a form on the machine. The application program (AP) that implements the withdrawal transaction issues read and/or write commands to a file manager to access files or uses a DBMS query language to access databases. The user is unaware of the actions of the underlying TPS. A transaction run on behalf of a user normally completes successfully, but if it does not, then a message is issued to the user describing the problem and the TPS aborts the transaction. It appears to the user that he is the only one using the TPS.

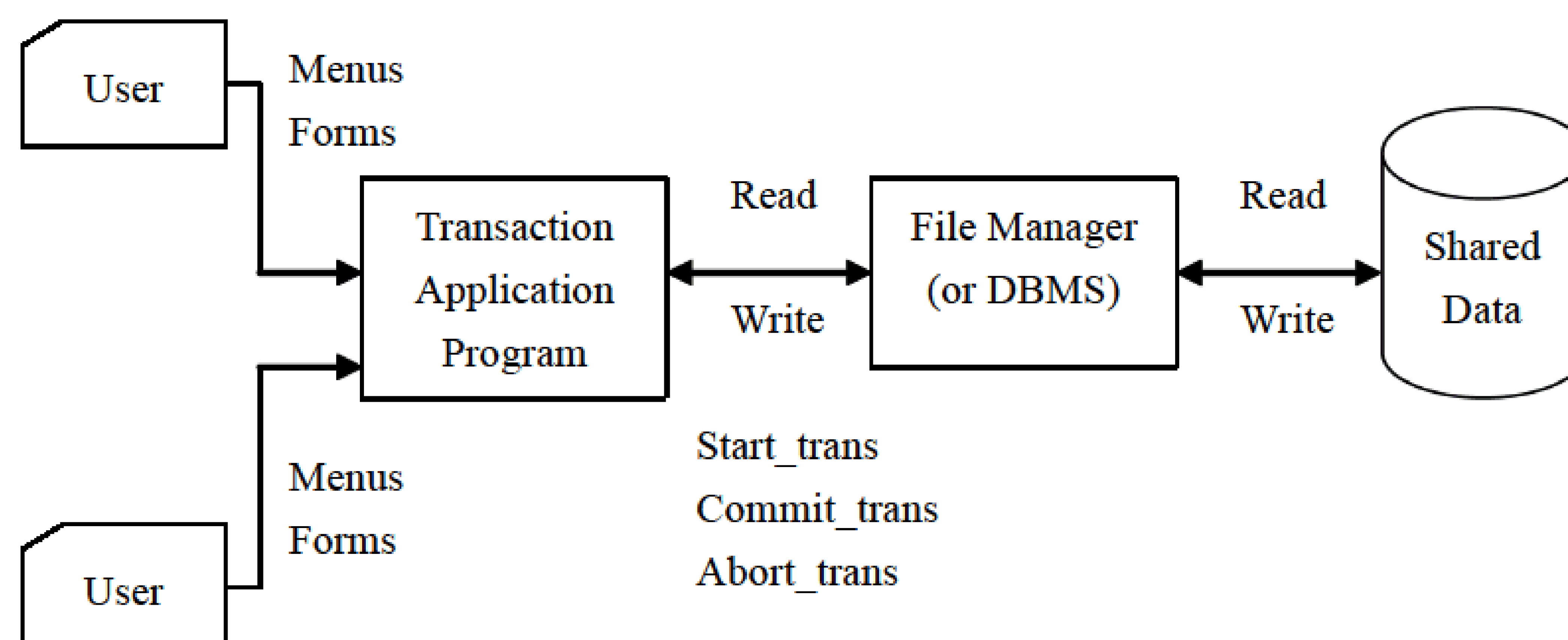


Figure 15-2　User's/Programmer's View of A TPS

The programmer's view is that his or her program will be the only one running on the system during its execution lifetime. He or she should not have to worry about the location of data when writing the program, and depends on the TPS to protect the program from update conflicts and from system failures.

The programmer defines the scope of a transaction in an AP by enclosing a sequence of programming language statements inside a pair of start_trans, commit_trans statements. Since many users may be simultaneously exercising the same transaction, the TP subsystem must be able to keep track of which instantiation of the transaction belongs to which user.

The implementer's view of a TPS is much different from the other views (see Figure 15-3). The implementer sees the system as a large number of interactive users accessing shared databases. The implementer's view is that of a transaction making its way through the system, that is, gaining

read and/or write locks on each record accessed and logging updates to records written. The access methods access data from shared files only after the access request is granted by the scheduler. A scheduler prevents concurrent accesses to data when conflicts occur. The access method routines not only invoke the scheduler to determine whether they can "safely" access the data, but they also invoke log manager routines to log redo and possibly undo data. The undo data is used by the recovery manager to abort transactions (that is, rollback transactions) and the recovery manager uses the redo data to recover after system crashes (undo data may also be required during recovery).

The operator sees the system as a collection of nodes, users, communications lines, applications, discs, archive tapes, procedures for recovering from system and media failures, and so forth. It is interesting that operators are the source of more system failures due to human errors than any other source. Regardless, operators are vital to the smooth operation of an online transaction processing (OLTP) system.

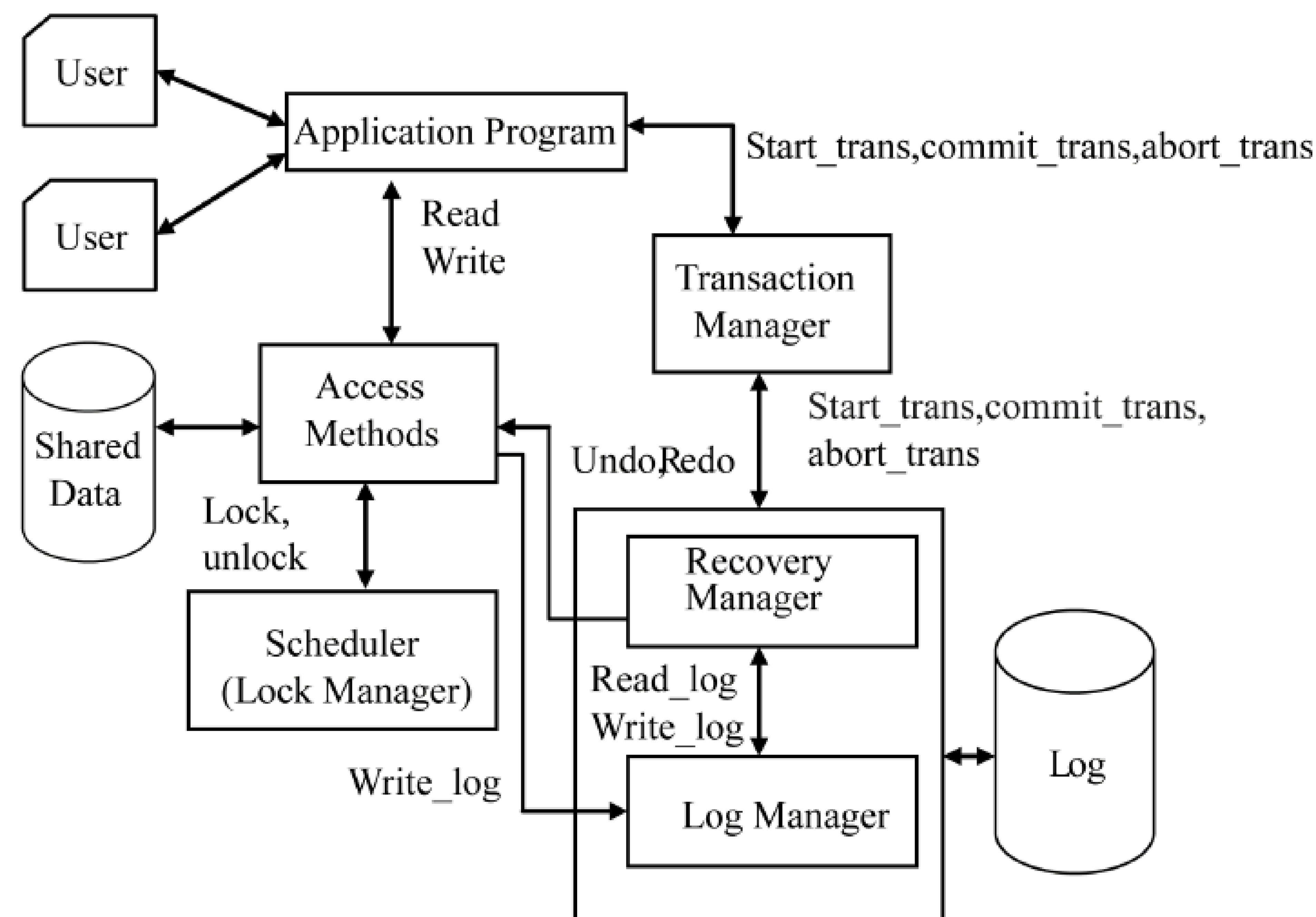


Figure 15-3 File Manager with Embedded TP Subsystem

15.3 Management Information System (MIS)

15.3.1 A Definition of MIS

We define a management information system (MIS) as a computer-based system that makes information available to users with similar needs. The users typically comprise a formal organizational chilly — the firm or a subsidiary subunit. The information describes the firm or one of its major system in terms of what has happened in the past, what is happening now, and what is likely to happen in the future. The information is made available in the form of periodic reports, special reports, and outputs of mathematical simulations. The information output is used by both managers and non-managers as they make decisions to solve the firm's problems.

15.3.2　A MIS Model

Our definition can be illustrated with the MIS model in Figure 15-4. The database contains the data provided by the AIS. In addition, both data and information are entered from the environment. The database contents are used by software that produces periodic and special reports, as well as mathematical models that simulate various aspects of the firm's operations. The software outputs are used by persons who are responsible for solving the firm's problems. Note that some of the problem solvers can exist within the firm's environment. The environment becomes involved when the firm bands together with such other organizations as suppliers to form an inter-organizational information system (IOS). In that case, the MIS supplies information to the other members of the IOS.

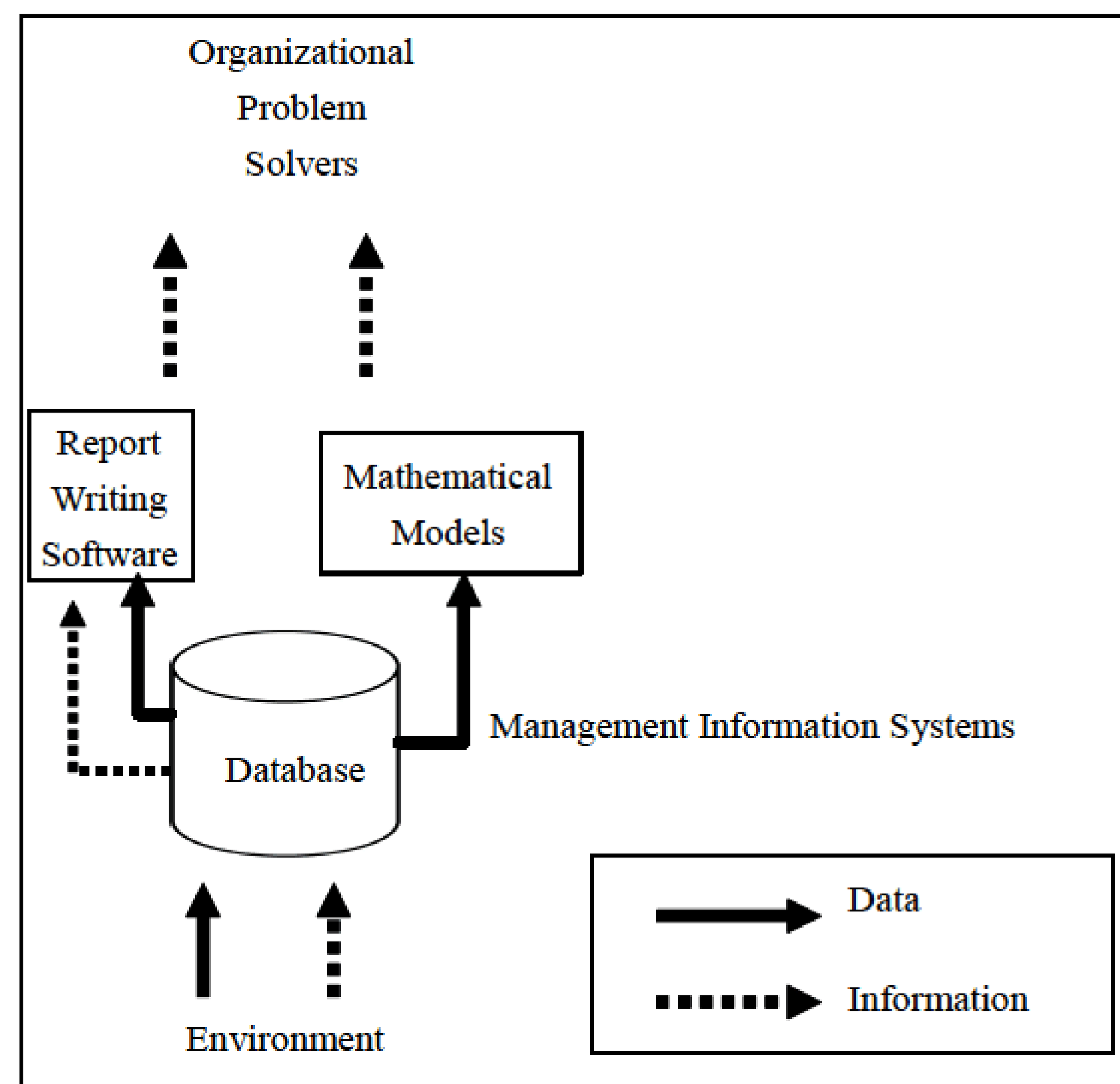


Figure 15-4　MIS Model

15.3.3　Organizational Information System

As firms gained experience in implementing companywide MIS designs, managers in certain areas began applying the concept to their own needs. These functional information systems, subsets of the MIS tailored to meet users' needs for information concerning functional areas, received much publicity in some areas. Marketing was the first area to conceive the notion of a functional information system. Considerable effort went into describing how the computer could be applied to the entire range of marketing operations.

The manufacturing area also embraced computer processing and applied the technology as both a conceptual information system and as a component in the physical manufacturing system. Technologies such as factory robots and computer-controlled conveyor systems are examples of how physical manufacturing processes have been computerized. Efforts to apply the computer as a conceptual system usually did not come under the name of manufacturing information systems. Rather, such names as material requirements planning (MRP) and computer integrated manufacturing (CIM) were adopted. We group all of these applications under the name of manufacturing information systems. Similarly, the financial area did not embrace the title of financial information systems, although computer-based information systems in that area are commonplace.

A functional area that did adopt the information system title was human resources, the area where most current attention is focused. The terms human resources information system (HRIS) and human resource management system (HRMS) are common.

Much current attention is also being focused on an organizational information system aimed at a management level — the executive information system, or EIS.

A strange phenomenon is the fact that the information services unit has demonstrated no real interest in applying the MIS concept to its own area, a situation akin to the shoemaker's children going barefoot. The chief information officer (CIO) and other information system (IS) managers have made use of computer-generated information over the years but have not formalized such use in an information system.

Figure 15-5 illustrates the subdivision of the MIS into organizational subsystems. Although the figure shows clear lines of separation, there is nothing physically separating them. Much of the database that is used by one organizational subsystem can also be used by others, and there may be much sharing of software. Organizational information systems are a logical rather than a physical way of thinking about the MIS.

We have used the term computer-based information system to represent all the firm's information systems. A term that is currently very popular is enterprise information system. It can perform all the firm's computer processes, from data processing tasks to the preparation of management information. The term enterprise resource planning (ERP) describes the manner in which the enterprise information system is developed. ERP is an extension of the MRP concept developed in manufacturing to encompass the entire firm.

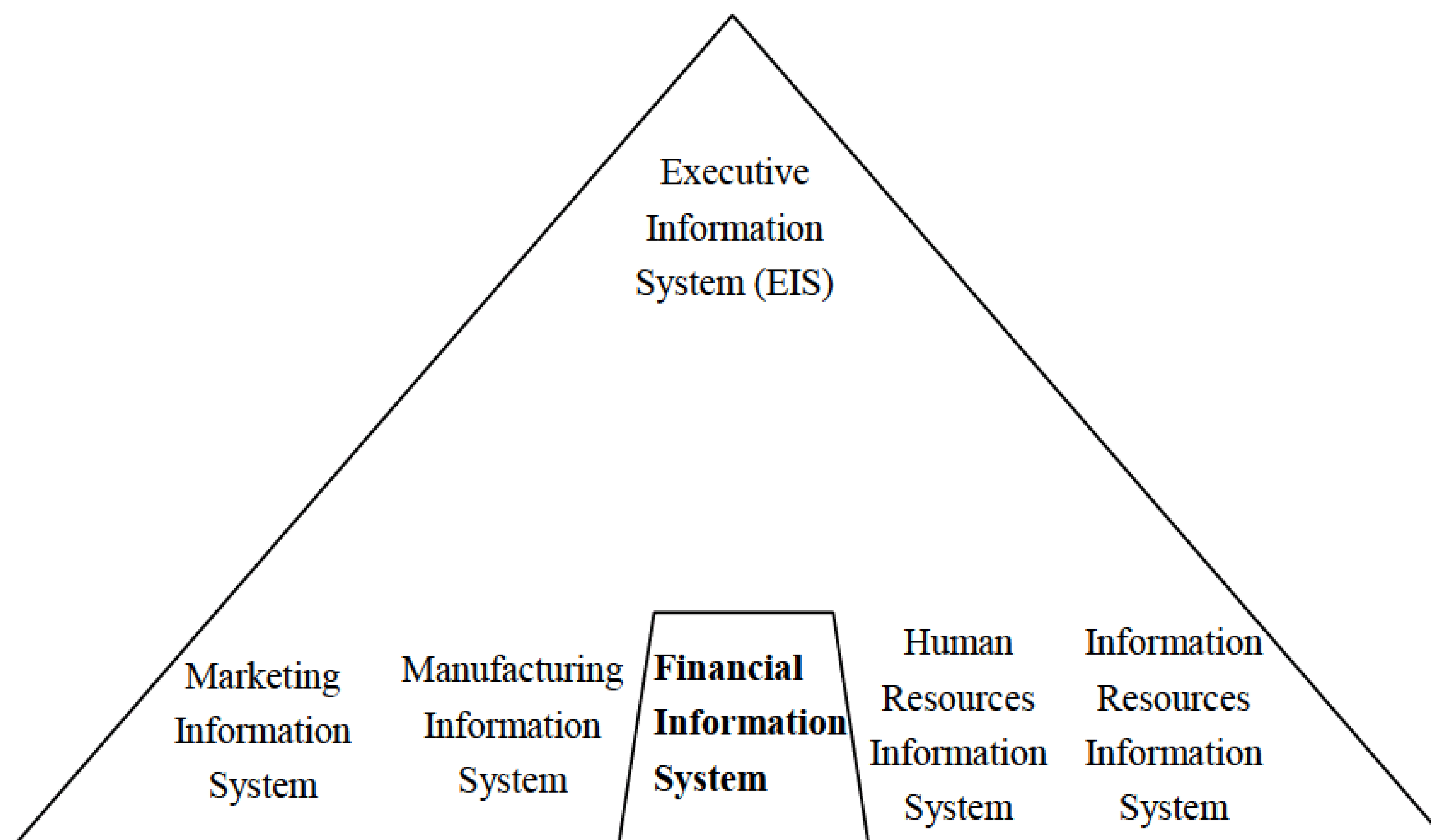


Figure 15-5 Organizational Information Systems

15.4 Decision Support System (DSS)

15.4.1 The Concept of DSS

The Concept of DSS is originated in the late 1960s with computer timesharing. For the first time, a person could interact directly with the computer without having to go through information specialists.

It was not until 1971, however, that the term DSS was coined by G. Anthony and Michael S. Scott Morton, both MIT professors. They felt a need for a framework to channel computer applications toward management decision making, and they developed what has become known as the Gorry and Scott Morton grid. The grid, illustrated in Figure 15-6, is based on both Simon's concept of programmed and non-programmed decisions and management-theorist Robert N. Anthony's concept of management levels.

Gorry and Scott Morton described decision types in terms of problem structure, ranging from structured to semistructured to unstructured.

Simon's phases of decision making are used to determine the structure of managerial problems. A fully structured problem is one in which the first three of Simon's phases — intelligence, design, and choice — are structured. That is, it is possible to specify algorithms, or decision rules, that allow the problem to be identified and understood, alternative solutions to be identified and evaluated, and a solution to be selected. An unstructured problem, on the other hand, is one in which none of the three phases is structured. A semistructured problem is one in which one or two of the phases are structured.

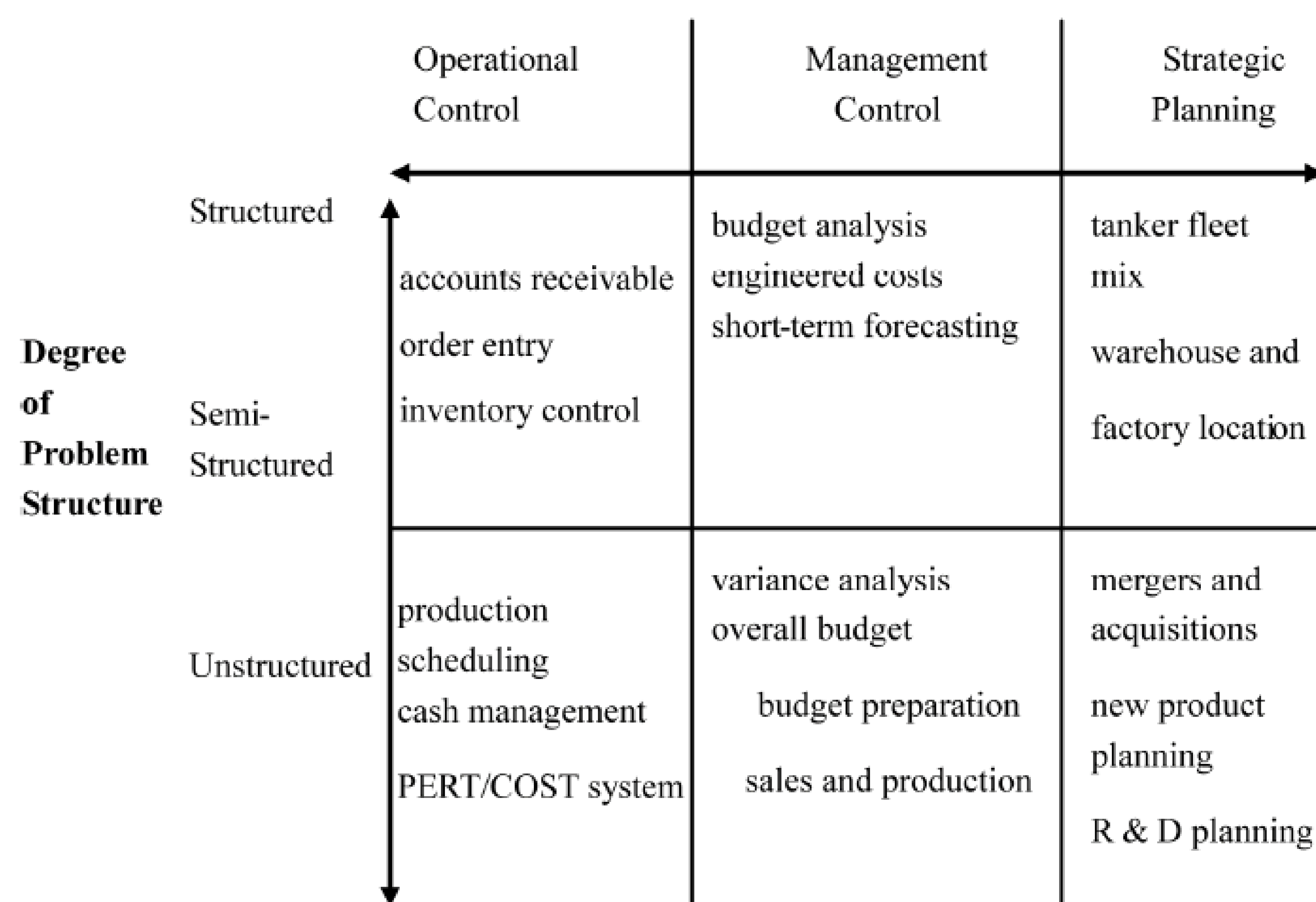


Figure 15-6　the Gorry and Scott Morton Grid

Gorry and Scott Morton entered types of business problems into their grid. As an example, accounts receivable planning is accomplished by managers on the operational control level, who make structured decisions. R & D planning is accomplished by strategic planning managers making unstructured decisions.

The horizontal dotted line through the middle of the grid is significant. It separates the problems that had, at that time, been successfully solved with computer assistance (above) from those problems that had not been subjected to computer processing. The upper area was named structured decision systems (SDS), and the lower area was named decision support systems (DSS).

Gorry and Scott Morton initially regarded DSS as describing only future computer applications. Subsequently, the term has been applied to all computer applications dedicated to

decision support — both current and future.

15.4.2 DSS Objectives

A DSS should achieve three objectives.

- Assist managers in making decisions to solve semistructured problems.
- Support the manager's judgment rather than try to replace it.
- Improve the manager's decision-making effectiveness rather than its efficiency.

These objectives correlate with three fundamental principles of the DSS concepts: problem structure, decision support, and decision effectiveness.

1. Problem Structure

It is difficult to find problems that are completely structured or unstructured. The vast majority are semistructured — Simon's gray area. This means that the DSS is aimed at the area where most problems are found.

2. Decision Support

The DSS is not intended to replace the manager. The computer can be applied to the structured portion of the problem, but the manager is responsible for the unstructured portion — applying judgment or intuition and conducting analyses. The manager and the computer work together as a problem-solving team in solving problems that fall in the large semistructured area.

3. Decision Effectiveness

The objective of the DSS is not to make the decision-making process as efficient as possible. The manager's time is valuable and should not be wasted, but the main benefit of using a DSS is better decisions.

When making a decision, the manager does not always try to achieve the best one. Some mathematical models will do that for the manager. However, in most cases it is the manager who must decide which alternative is best. Quite possibly the manager could spend extra time to fine tune the solution so that it approaches the optimum, but the increased precision would not be worth the time and effort. The manager uses judgment to determine when a decision will contribute to a problem solution.

15.4.3 A DSS Model

As shown in Figure 15-7, data and information are entered into the database from the firm's environment. The database also contains data that is provided by the AIS. The contents of the database are used by three software subsystems.

- Report-writing software produces both periodic and special reports. Periodic reports are prepared according to a schedule, and typically they are produced by software that is coded in such a procedural language as COBOL or PL/1. The special reports are prepared in response to unanticipated information needs and take the form of database queries by users who use the query language of a DBMS or a fourth-generation language.

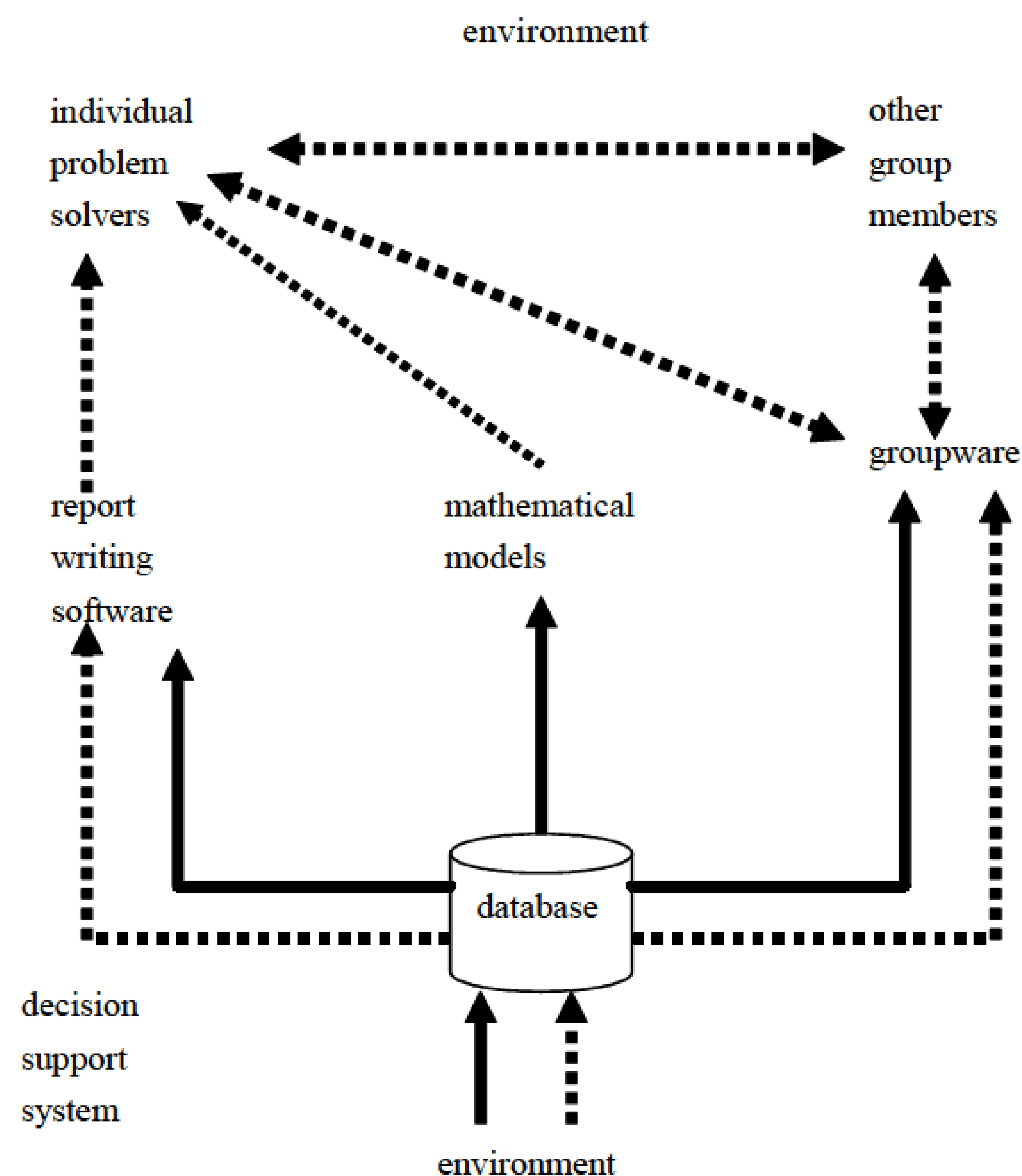


Figure 15-7　A Model of Decision Support System

- Mathematical models produce information as a result of either simulations that involve one or more components of the physical system of the firm or facets of its operations. Mathematical models can be written in any procedural programming language. However, special modeling languages make the task easier and have the potential of doing a better job.
- Groupware enables multiple problem solvers, working together as a group, to reach solutions. In this particular situation, the term GDSS, or group decision support system is used. Perhaps the problem solvers represent a committee or a project team. The group members communicate with one another both directly and by means of the groupware.

The report-writing software and mathematical models have always been regarded as necessary DSS ingredients. As the DSS concept was broadened to provide support to two or more problem solvers working together as a team or committee, the idea of special group-oriented software, or groupware, became a reality.

During the past decade, most of the DSS effort has been aimed at refining the group decision support system concept.

Technical Notes to the Text

1. transaction processing systems (TPS)，事务处理系统。TPS 是使操作层的数据处理自动化，提高工作效率的信息系统。事务处理是组织中最基本的活动，把信息按照事务组织和处理，实现组织日常事务活动的系统称为事务处理系统，其主要作用是反馈控制。

2. online transaction processing (OLTP)，联机事务处理。联机事务处理的对象是大量的事

务，每个事务中有相对小容量的细节数据，而 OLAP 系统则注重于对相对大容量的、主要是聚合的数据进行分析。其基本特征是综合用户数据进行动态实时多维分析，提供给用户快速一致的查询响应速度。

3. management information system (MIS)，管理信息系统。管理信息系统是一个由人、计算机等组成的，能进行管理信息的收集、传递、存储、加工、维护和使用的系统。

4. inter-organization information systems (IOS)，组织间信息系统。组织间信息系统使组织间的信息自由流动，用于支持产品和服务的计划、设计、开发、生产和交付。

5. material requirements planning (MRP)，物料需求计划。MRP 是将企业主生产计划的产品级计划通过网络规划的方法，按照物料清单分解为零件级的计划，并直接指挥着自制件的生产数量和进度，以及外购件的数量和配套日期。

6. computer integrated manufacturing system (CIMS)，计算机集成制造系统。CIMS 是一种组织、管理和运行现代制造类企业的理念。它将传统的制造技术与现代信息技术、管理技术、自动化技术、系统工程技术等有机结合，使企业产品整个生命周期的各阶段活动中有关的人/组织、经营管理和技术三要素及其信息流、物流和价值流三者有机集成并优化运行，以达到产品(Product)上市快、交货期(Time)短、质量(Quality)高、成本(Cost)低、服务(Service)好、环境(Environment)好，进而提高企业的柔性、健壮性、敏捷性，使企业赢得市场竞争。

7. executive information system (EIS)，主管信息系统。EIS 是高度交互的 MIS 与 DSS 和人工智能系统的结合体。简而言之，EIS 既具有 MIS 的功能，又具有 DSS 的功能。

8. structured problem，结构化问题。结构化问题指决策是重复性的和常规性的，并且在处理这种决策时有一套确定的方法。

9. semistructured problem，半结构化问题。半结构化问题指介于结构化决策和非结构化决策之间的决策问题。

10. unstructured problem，非结构化问题。非结构化问题指不能用常规的方法来处理、无任何规律可循的决策。

11. decision support system (DSS)，决策支持系统。决策支持系统指能够利用数据和模型来帮助决策者解决非结构化问题的、高度灵活的人机交互式计算机信息系统。其目的是支持解决非结构化的决策问题，进一步提高决策的效果。DSS 的一个典型例子是电子报表(spreadsheet)。利用 DSS 可以进行各种数据分析，如 What-if 分析、回归分析等。

12. group decision support system (GDSS)，群体决策支持系统。群体决策支持系统是一种在 DSS 基础上利用计算机网络与通信技术，供多个决策者为了一个共同的目标，通过某种规程相互协作地探寻半结构化或非结构化决策问题解决方案的信息系统。

13. strategy support systems (SSS)，战略支持系统。战略支持系统是指由支持战略目标实现的若干管理重点课题构成的各分项系统的集合，主要分为职能支持系统、阶段支持系统。职能支持系统是由企业各职能单位的管理重点课题构成的系统集合，主要包括人力资源系统、财务系统、生产系统、营销系统等。阶段支持系统是由各战略阶段的管理重点课题构成的系统集合。随着竞争的加剧，企业管理不可控因素越来越多，为了提高战略调控能力，确保战略目标的顺利实现，企业战略历程可以划分成若干个战略阶段来进行。

14. chief information officer (CIO)，首席信息官，或称为信息主管。根据 Gartner Group 的定义，CIO 是一个机构关键的管理岗位，负责制定信息化政策，保证 IT 和业务发展战略的

默契配合。综合来说，CIO 的主要职责包括以下 10 个方面：①参与制定组织发展战略，领导组织信息战略的制定。②确立信息处理和利用及其所需设备方面的政策、标准和程序，制定组织信息制度和信息政策。③培育良好的信息文化。④提升组织和员工的信息素质、信息能力。⑤为高层管理者提供决策所需的信息支持和信息能力支持。⑥进行信息化项目规划，领导重要信息化项目的实施。⑦监控所有信息化项目的实施及现有信息系统的运行。⑧领导组织内所有信息部门为操作部门和业务功能提供咨询或服务。⑨与业务部门一起考虑如何使信息和知识为产品或服务增值。⑩将自己的经验和教训等知识贡献给行业协会和社会。

Word Bank to the Text

A. Useful new words

query	*n*. 质问，询问
abort	*v*. 异常中断
execution	*n*. 实行，执行
enclose	*v*. 放入封套，装入
instantiation	*n*. 实例化，例示
archive	*v*. 存档　*n*. 档案文件
regardless	*adj*. 不管，不顾
comprise	*v*. 包括，包含
subsidiary	*adj*. 辅助的，补充的
simulation	*n*. 仿真，假装
conceive	*v*. 构思，考虑
embrace	*v*. 拥抱，互相拥抱，包含
commonplace	*adj*. 普通的，平凡的
extension	*n*. 延伸，伸展
encompass	*v*. 包围，围绕
horizontal	*adj*. 地平线的，水平的
effectiveness	*n*. 效力
intuition	*n*. 直觉
finetune	*v*. 调整
refine	*v*. 精炼，精制
interrelate	*v*. 互相联系

B. Useful expressions

during one’s lifetime	一生中，存在期
keep track of	追踪
make one’s way through	穿越
recover from	恢复

and so forth	等等
be involved in	涉及，参与
together with	和，连同
gain experience in	获取经验
meet users' need	满足用户需求
under the name of	以……的名义
focus attention on	关注
be akin to	同类的，相似的
range from … to	在……范围内变化
be subjected to	遵照，服从
in most cases	在多数情况下
be worth the time and effort	值得付出时间和努力
in response to	响应，适应
reach solutions	得到解决办法
by means of	通过，借助于

C. Technical terms and proper names

transaction processing system (TPS)	事务处理系统
concurrency control	并发控制
recovery	恢复
functional information system	职能信息系统
management information system (MIS)	管理信息系统
inter-organizational information system (IOS)	组织间信息系统
material requirements planning (MRP)	物料需求计划
computer integrated manufacturing (CIM)	计算机集成制造
manufacturing information system	制造业信息系统
human resources information system (HRIS)	人力资源信息系统
human resource management system (HRMS)	人力资源管理系统
strategy information system (SSS)	战略支持系统
accounting information system (AIS)	会计信息系统
static or dynamic model	静态或动态模型
probabilistic or deterministic model	概率或确定性模型
optimizing or sub-optimizing model	最优化或局部最优化模型
structured problem	结构化问题
semistructured problem	半结构化问题
unstructured problem	非结构化问题
strategic planning	战略计划
management control	管理控制
operational control	运行控制

structured decision system (SDS)　　结构化决策系统
decision support system (DSS)　　决策支持系统
group decision support system (GDSS)　　群体决策支持系统

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. A DSS should assist managers in making decisions to solve _______ problems.
2. The _______ sees TPS as a large number of interactive users accessing shared databases.
3. In DSS, _______ produce information as a result of either simulations that involve one or more.
4. A DSS model includes: _______, mathematical models, and _______.
5. One objective of a DSS is improving the manager's decision-making _______ rather than its _______.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The objective of the DSS is to make the decision-making process as efficient as possible. (　　)
2. DSS should assist managers in making decisions to solve structured decision problems. (　　)
3. The DSS is aimed at the area where most problems are found. (　　)
4. The implementer sees TPS as a large number of interactive users accessing shared databases. (　　)
5. In DSS, groupware enables multiple problem solvers, working together as a group, to reach solutions. (　　)

Ⅲ. Match each of the following terms with the appropriate definition.

transaction	MIS	EIS	HRIS

1. _______ An atomic unit of execution that manipulates system resources.
2. _______ Executive information system.
3. _______ A computer-based system that makes information available to users with similar needs.
4. _______ Human resources information system.

Ⅳ. Point out which is structured problem, semi-structured and unstructured problem.

1. Order entry　　2. Cash management
3. PERT/COST system　　4. Short-term forecasting
5. Sales and production　　6. R&D planning

7. New product planning
8. Budget preparation
9. Accounts receivable
10. Inventory control

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

refine	embrace	archive
conduct	horizontal	regardless
intuition	Comprise	abort

1. When the decision was made to ________ the mission, there was great confusion.
2. I decided I would go to the ________ the next day and look up the appropriate issue.
3. ________ of whether he is right or wrong, we have to abide by his decisions.
4. The task force is ________ of congressional leaders, cabinet heads and administration officials.
5. The new rules have been ________ by government watchdog organizations.
6. The board consists of vertical and _______ lines.
7. Nobody told me where to find you. It was sheer ________.
8. Most people believe they ________ their private and public lives in accordance with Christian morality.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "grant" and fill in the sentences with the right word.

Example: The access methods access data from shared files only after the access **request** is **granted** by the scheduler.

1. It was a Labour government which granted ________ to India and Pakistan.
2. The US agreed to grant the new state diplomatic ________.
3. As long as the students pass final exams, American authorities grant a ________, which is recognized in China.
4. Only after several months did officials grant ________ to visit his sister in jail.
5. The minister granted journalists an ________.
6. The old lady granted her all ________ to her only heir, a lovely pet cat.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

An Information System is the system of persons, data records and activities that process the data and information in a given organization, __1__ manual processes or automated processes. The computer-based information systems are the __2__ of study for information technologies (IT).

Information systems deal with the development, use and management of an organization's IT infrastructure. In the __3__ -industrial, information age, the focus of companies has shifted __4__ being product oriented to knowledge __5__, in a sense that market operators today compete on process and innovation rather than product: the emphasis has shifted from the quality and quantity of production, to the production process __6__, and the services that __7__ the production process.

The biggest __8__ of companies today, is their information, represented in people, experience, know-how, innovations (patents, copyrights, trade secrets), and __9__ a market operator to be able to compete, he/she must have a strong information infrastructure, at the heart of __10__, lies the information technology infrastructure. Thus, the study of information systems focuses on why and how technology can be put into best use to serve the information flow within an organization.

1. A. including	B. include	C. included	D. includes
2. A. subject	B. course	C. field	D. arts
3. A. post	B. after	C. now	D. present
4. A. to	B. from	C. with	D. within
5. A. directed	B. objected	C. named	D. oriented
6. A. himself	B. itself	C. themselves	D. herself
7. A. company	B. accompany	C. companion	D. accompanies
8. A. task	B. object	C. asset	D. plan
9. A. for	B. of	C. as	D. with
10. A. that	B. whoever	C. what	D. which

Translation

Ⅷ. Translate the following into Chinese.

1. Since many users may be simultaneously exercising the same transaction, the TP subsystem must be able to keep track of which instantiation of the transaction belongs to which user.

2. The access method routines not only invoke the scheduler to determine whether they can "safely" access the data, but they also invoke log manager routines to log redo and possibly undo data.

3. Another characteristic of transaction processing mentioned earlier is that once a transaction commits, the data updated by it is persistent.

4. The information specialists decided that the only solution was for them to design and implement systems to produce information that they thought the managers needed.

5. The horizontal dotted line separates the problems that had, at that time, been successfully solved with computer assistance from those problems that had not been subjected to computer processing.

Chapter 16
Enterprise Resource Planning

Pre-reading Questions

1. What is an enterprise resource planning system?
2. Describe the evolution of enterprise resource planning system.
3. Describe the characteristics and core functions found in the successful ERP system.

16.1 Enterprise System

There's a lot of sloppy terminology flying around today in the business press, and one misnomer is to label enterprise-wide transaction processing software systems as ERP. These software packages support effective resource planning and make much of it feasible, but they don't truly do it.

Therefore, we need to trot out another acronym that does refer to software: ES. This stands for Enterprise System or Enterprise Software. In his book Mission Critical, author Thomas H. Davenport describes enterprise systems as "packages of computer applications that support many, even most, aspects of a company's information needs."

An enterprise system (ES) is a large software application that companies use to manage their operations and to distribute information of all kinds to their workforce, to their suppliers, and to their customers. From finance to manufacturing, from sales to services, enterprise systems support thousands of business activities.

That makes sense to us. Now for another distinction: not all ERP business functions are contained in the typical enterprise software (ES) suite. Similarly, the typical ES contains software support for business processes that are not part of ERP. In Figure 16-1, we can see that distinction graphically. Please note the three areas on that diagram. The rightmost part of the figure refers to those functions contained within a typical ES that are not parts of ERP; the leftmost area refers to those ERP functions not normally supported by an ES; the area of overlap in the center references those ERP functions typically supported by enterprise software.

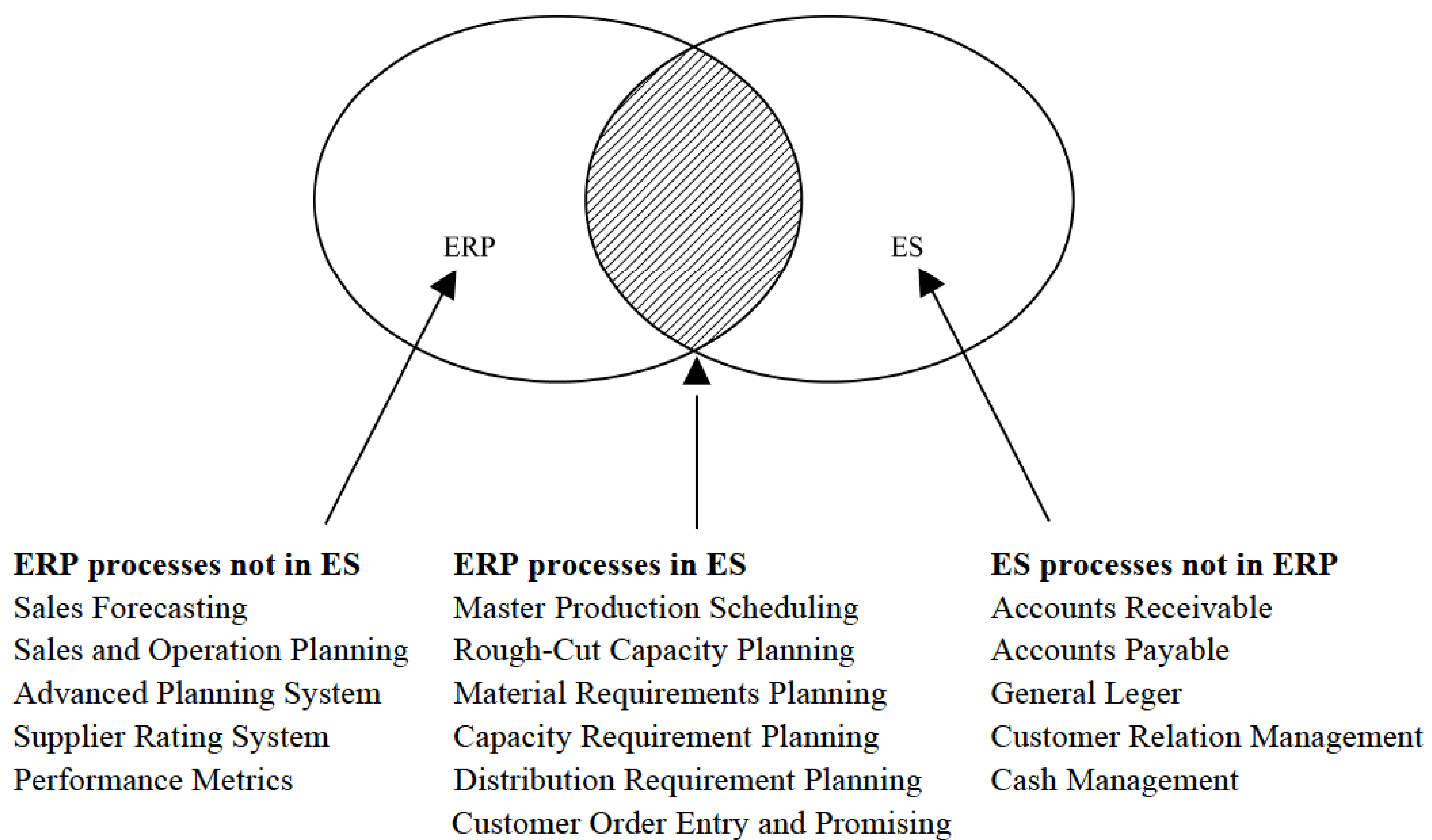


Figure 16-1　The Relationship and Difference of ERP and ES

Now, let's take a look at just what this ERP thing is all about.

16.2　Enterprise Resource Planning System

Enterprise Resource Planning (ERP) — and its predecessor, Manufacturing Resource Planning (MRP II) — is helping to transform our industrial landscape. It's making possible profound improvements in the way manufacturing companies are managed. It is a strong contributor to America's amazing economic performance of the 1990s and the emergence of the New Economy. A half century from now, when the definitive industrial history of the twentieth century is written, the evolution of ERP will be viewed as a watershed event. Let's describe Enterprise Resource Planning as:

An enterprise-wide set of management tools that balances demand and supply, containing the ability to link customers and suppliers into a complete supply chain, employing proven business processes for decision-making, and providing high degrees of cross-functional integration among sales, marketing, manufacturing, operations, logistics, purchasing, finance, new product development, and human resources, thereby enabling people to run their business with high levels of customer service and productivity, and simultaneously lower costs and inventories; and providing the foundation for effective e-commerce.

Here are some descriptions of ERP, not definitions but certainly good examples.

Enterprise resource planning is a company increasing its sales by 20 percent in the face of an overall industry decline. Discussing how this happened, the vice president of sales explained, "We're capturing lots of business from our competitors. We can out-deliver them. Thanks to ERP, we can now ship quicker than our competition, and we ship on time."

Enterprise resource planning is a Fortune 50 corporation achieving enormous cost savings and acquiring a significant competitive advantage. The vice president of logistics stated: "ERP has provided the key to becoming a truly global company. Decisions can be made with accurate data and with a process that connects demand and supply across borders and oceans. This change is worth billions to us in sales worldwide."

Enterprise resource planning is a purchasing department generating enormous cost reductions while at the same time increasing its ability to truly partner with its suppliers. The director of purchasing claimed, "For the first time ever, we have a good handle on our future requirements for components and raw materials. When our customer demand changes, we and our suppliers, can manage changes to our schedules on a very coordinated and controlled basis. I don't see how many company can do effective supply chain management without ERP."

American Production and Inventory Control Society (APICS) has defined an ERP system as "a method for the effective planning and controlling of all the resources needed to take, make, ship, and account for customer orders in a manufacturing, distribution or service company". ERP systems are commercial software that provides the seamless integration of all the information flowing through the company — financial, accounting, human resources, supply chain, and customer information. ERP systems are configurable information systems packages that integrate information and information-based processes within and across functional areas in an organization. What is really means is one database, one application, and a unified interface across the enterprise. An ERP system is required to have the following characteristics: (1) modular design comprising many distinct business functions such as financial, manufacturing, distribution, and the like, (2) centralized DBMS, (3) integrated functions that provide seamless information flow among the functions, (4) flexible, best business practices, (5) functions that work in real time, and (6) Internet-enabled.

Different ERP vendors provide ERP systems with some degree of specialty but the core functions are almost the same for all of them. Some of the core ERP functions found in the successful ERP systems include: accounting management, financial management, manufacturing management, production management, transportation management, sales and distribution management, human resources management, supply chain management, customer relationship management, and E-business.

The functions of an ERP system can either work as standalone units, or several functions can be combined together to form an integrated system.

You need to realize that ERP systems will not improve organizations' functionalities overnight. The high expectation of achieving cost savings and service improvements is very much dependent on how good the chosen ERP system fits the organizational functionalities and how well the tailoring and configuration process of the system matches with the business culture, strategy, and structure of the organization. Overall, an ERP system is expected to improve both back-office and front-office functions simultaneously. Organizations choose and deploy ERP systems for many different benefits and reasons. In many cases the calculation of return on investment (ROI) is

weighted against the many benefits. The benefits that an ERP system may bring to organizations are described in Table 16-1. To reap the benefits of ERP systems, however, organizations need to overcome certain problems and disadvantages, which are also listed in Table 16-1.

Table 16-1 Advantages and Disadvantages of Enterprise Resource Planning Systems

Advantages of ERP systems	
Advantage	Benefit
reliable information access	common DBMS, consistent and accurate data, improved reports
avoid data and operations redundancy	modules access same data from the central database, avoid multiple data input, and output operations
delivery and cycle time reduction	minimizes retrieving and reporting delays
cost reduction	time savings, improved control by enterprise-wide analysis of organizational decisions
easy adaptability	changes in business processes easy to adapt and restructure
improved scalability	structured and modular design with "add-ons"
improved maintenance	vendor-supported long-term contract as part of the system procurement
global outreach	extended modules such as CRM and SCM
e-business	internet commerce, collaborative culture
Disadvantages of ERP Systems	
Disadvantage	How to overcome
time-consuming	minimize sensitive issues, internal politics, and raise general consensus
expensive	cost may vary from thousands of dollars to millions; business process reengineering (BPR) cost may be extremely high — be cautious
lack of conformity of the modules	the architecture and components of the selected system should conform to the business processes, culture, and strategic goals of the organization
vendor dependence	single vendors vs. multivendor consideration, options for "best of breeds", look for long-term committed support
too many features, too much completeness	ERP system may have too many features and modules so the user needs to consider carefully and implement the needed only
Disadvantages of ERP Systems	
Disadvantage	How to overcome
questionable scalability and global outreach	look for vendor investment in R & D, long-term committed to product and services, consider Internet-enabled systems
not enough extended ERP capability	be sure "add-on" facilities and extended modules such as CRM and SCM are available

16.3 The Evolution of Enterprise Resource Planning

ERP system replace "islands of information" with a single, packaged software solution that

integrates all the traditional enterprise management functions such as financials, human resources, and manufacturing and logistic. Knowing the history and evolution of ERP is essential to understand its current application and its future developments. To give you a better perspective, let's review the evolution of ERP systems.

Stage One —Material Requirements Planning (MRP)

ERP began life in the 1960s as Material Requirements Planning (MRP), an outgrowth of early efforts in bill of material processing. MRP's inventors were looking for a better method of ordering material and components, and they found it in this technique. The logic of material requirements planning asks the following questions:

- What are we going to make?
- What does it take to make it?
- What did we have?
- What do we have to get?

This called the universal manufacturing equation. Material Requirements Planning system simulates the universal manufacturing equation. It uses the master schedule (What are we going to make?), the bill of material (What does it take to make it?), and inventory records (What did we have?) to determine future requirements (What do we have to get?).

For a visual depiction of this and the subsequent evolutionary steps, please see Figure 16-2, a modified version of a diagram in Carol Ptak's recent book on ERP.

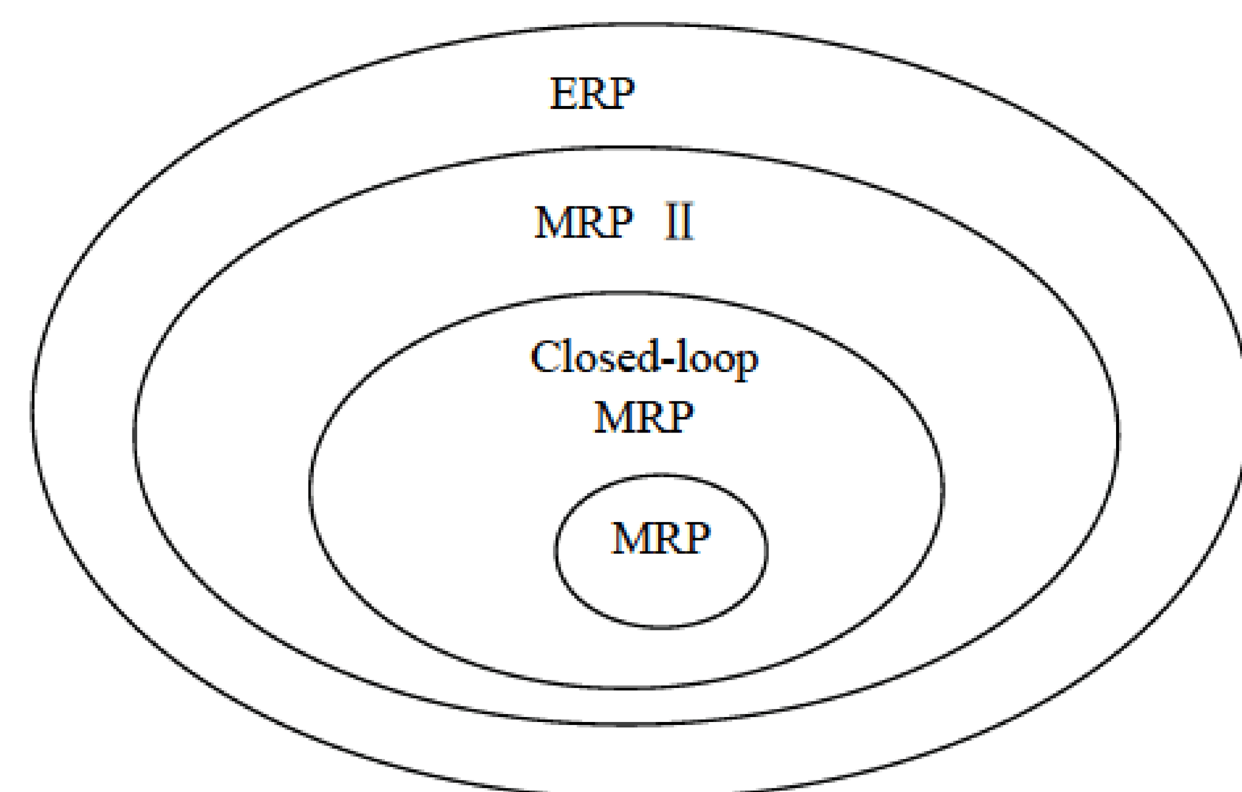


Figure 16-2 Evolution of ERP

Stage Two — Closed-Loop MRP

MRP quickly evolved, however, into something more than merely a better way to order. Early users soon found that Material Requirements Planning contained capabilities far greater than merely giving better signals of reordering. They learned this technique, so they could help to keep order due dates valid after the orders had been released to production or to suppliers. MRP could detect when the due date of an order (when it's scheduled to arrive) was out of phase with its need date (when it's required).

This was a breakthrough. For the first time ever in manufacturing, there was a formal mechanism for keeping priorities valid in a constantly changing environment. This is important, because in a manufacturing enterprise, change is not simply a possibility or even a probability. It's a certainty, the only constant, the only sure thing. The function of keeping order due dates valid and synchronized with these changes is known as priority planning.

So, did this breakthrough regarding priorities solve all the problems? Was this all that was needed? The issue of priority is only half the battle. Another factor — capacity — represents an equally challenging problem (see Table 16-2).

Table 16-2 Priority vs. Capacity

Priority	Capacity
Which ones?	Enough?
Sequence	Volume
Scheduling	Loading

Techniques for helping plan capacity requirements were tied in with material requirements planning. Further, tools were developed to support the planning of aggregate sales and production levels (Sales & Operations Planning); the development of the specific build schedule (Master Scheduling); forecasting, sales planning, and customer-order promising (Demand Management); and high-level resource analysis (Rough-Cut Capacity Planning). These developments resulted in the second step in this evolution: closed-loop MRP (see Figure 16-3).

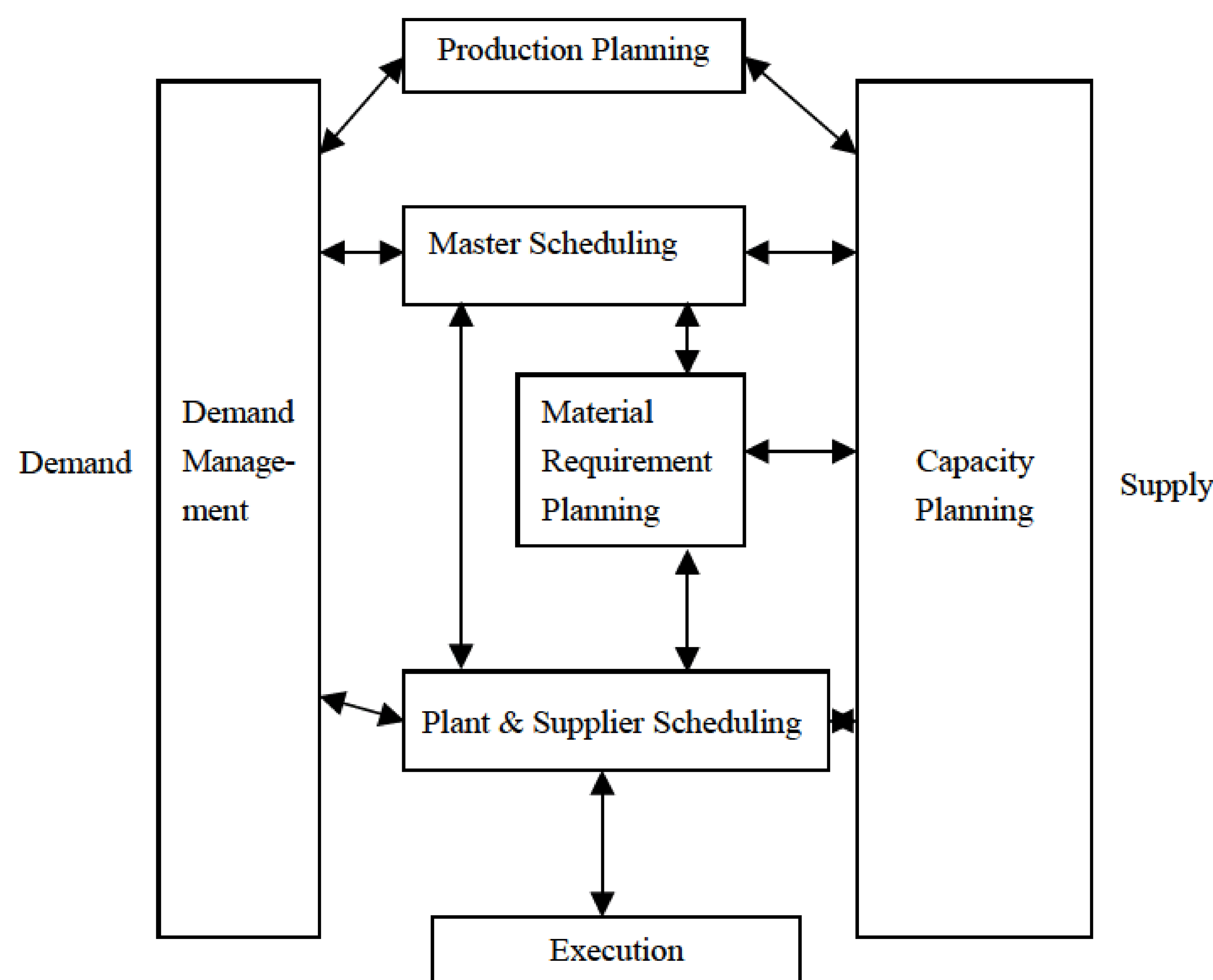


Figure 16-3 Closed-loop MRP

Closed-loop MRP has a number of important characteristics.

- It's a series of functions, not merely material requirements planning.
- It contains tools to address both priority and capacity, and to support both planning and execution.
- It has provisions for feedback from the execution functions back to the planning functions. Plans can then be altered when necessary, thereby keeping priorities valid as conditions change.

Stage Three — Manufacturing Resource Planning (MRP II)

The next stage in this evolution is called Manufacturing Resource Planning or MRP II (to distinguish it from Material Requirements Planning, MRP). As a direct outgrowth and extension of closed-loop MRP, it involves three additional elements.

- Sales & Operations Planning — a powerful process to balance demand and supply at the volume level, thereby providing top management with far greater control over operational aspects of the business.
- Financial interface — the ability to translate the operating plan (in pieces, pounds, gallons, or other units) into financial terms (dollars).
- Simulation — the ability to ask "what-if " question actionable answers— in both unit and dollars. Initially this was done only on an aggregate, "rough-cut" basis, but today's advanced planning systems (APS) enable effective simulation at very detailed levels.

Now it's time to define Manufacturing Resource Planning. This definition, and the one to follow, come from APICS — The Educational Society for Resource Management. APICS is the leading professional society in this field, and its dictionary has set the standard for terminology over the years.

Manufacturing Resource Planning (MRP Ⅱ) is a method for the effective planning of all resources of a manufacturing company. Ideally, it addresses operational planning in units, financial planning in dollars, and has a simulation capability to answer "what-if " questions. It is made up of a variety of functions, each linked together: business planning, sales and operations planning, production planning, master scheduling, material requirements planning, capacity requirements planning, and the execution support systems for capacity and material. Output from these systems is integrated with financial reports such as the business plan, purchase commitment report, shipping budget, and inventory projections in dollars. Manufacturing resource planning is a direct outgrowth and extension of closed-loop MRP.

Stage Four — Enterprise Resource Planning (ERP)

The latest stage in this evolution is Enterprise resource planning (ERP). The fundamentals of ERP are the same as with MRP Ⅱ. However, thanks for large measure to enterprise software, ERP as a set of business processes is broader in scope, and more effective in dealing with multiple business units. Financial integration is even stronger. Supply chain tools, supporting business across company boundaries, are more robust.

Let's now look at a complete definition of ERP based on the description we saw a few pages back:

Enterprise resource planning (ERP) predicts and balances demand and supply. It is an enterprise-wide set of forecasting, planning, and scheduling tools, which: Links customers and suppliers into a complete supply chain, Employs proven processes for decision-making and Coordinates sales marketing, operations, logistics, purchasing, finance, product development, and human resources.

Its goals include high levels of customer service, productivity, cost reduction, and inventory turnover, and it provides the foundation for effective supply chain management and e-commerce. It does this by developing plans and schedules so that the right resource — manpower, materials, machinery, and money — are available in the right amount when needed.

Enterprise resource planning is a direct outgrowth and extension of manufacturing resource planning and, as such, includes all of MRPII's capabilities. ERP is more powerful in that it:

- Applies a single set of resource planning tools across the entire enterprise.
- Provides real-time integration of sales, operating, and financial data.
- Connects resource planning approaches to the extended supply chain of customers and suppliers.

The primary purpose of implementing enterprise resource planning systems is to run the business, in a rapidly changing and highly competitive environment, far better than before.

Technical Notes to the Text

1. enterprise system or enterprise software (ES)，企业系统或企业软件。

2. enterprise resource planning (ERP)，企业资源规划。ERP 可以从管理思想、软件产品、管理系统 3 个层次定义：①ERP 是由美国著名的计算机技术咨询和评估集团 Garter Group Inc. 提出的一整套企业管理系统体系标准，其实质是在 MRPⅡ的基础上进一步发展而成的面向供应链(supply chain)的管理思想。②ERP 是综合应用了客户机/服务器体系、关系数据库结构、面向对象技术、图形用户界面、第 4 代语言(4GL)、网络通信等信息产业成果，以 ERP 管理思想为“灵魂”的软件产品。③ERP 是整合了企业管理理念、业务流程、基础数据、人力物力、计算机硬件和软件的企业资源管理系统。总之，ERP 是指企业通过建立完善的现代企业管理信息系统，对企业资源进行统一规划、管理，以达到加强企业成本控制、提高资金利用效果、重视企业现代管理、提高企业整体实力的目的。

3. manufacturing resource planning (MRPⅡ)，制造资源计划。制造资源计划是指企业对其生产系统和经营活动建立一种计划模型，并通过利用该模型对企业的制造资源和经营任务的需求进行平衡，从而保证企业目标的实现。

4. material requirements planning (MRP)，物料需求计划。MRP 是将企业主生产计划的产品级计划通过网络规划的方法，按照物料清单分解为零件级的计划，并直接指挥自制件的生产数量和进度，以及外购件的数量和配套日期。

5. just in time (JIT)，即时生产。即时生产指将必要的原材料、零部件以必要的数量在必要的时间，并且只将所需要的零部件和原材料，只以所需要的数量、时间送到特定的生产线。

6. customer relation management (CRM)，客户关系管理。CRM 是一种旨在改善企业与客户之间关系的新型管理机制，它实施于企业的市场营销、销售、服务和技术支持等与客户相关的领域。CRM 的目标是：一方面通过提供更快速和周到的优质服务吸引和保持更多的客户；另一方面通过对业务流程的全面管理来降低企业的成本。

7. business process reengineering (BPR)，业务流程重组。业务流程重组是对企业的业务流程进行根本性的再思考和彻底性的再设计，从而获得在成本、质量、服务和速度等方面业绩的显著改善，使企业能最大限度地适应以“顾客、竞争和变化”为特征的现代企业经营环境。

Word Bank to the Text

A. Useful new words

sloppy	*adj*. 松散的，粗糙的
terminology	*n*. 术语学，专门名词
misnomer	*n*. 用词不当
acronym	*n*. 首字母简略词
rightmost	*adj*. 最右边的
leftmost	*adj*. 最左边的
profound	*adj*. 深刻的，意义深远的
emergence	*n*. 出现
definitive	*adj*. 最后的，确定的
watershed	*n*. 分水岭
inventory	*n*. 详细目录，存货
configurable	*adj*. 结构的，可配置的
deploy	*v*. 展开，配置
functionality	*n*. 功能性，泛函性
outgrowth	*n*. 派出，结果
depiction	*n*. 描写，叙述
evolutionary	*adj*. 进化的
regarding	*prep*. 关于
promising	*adj*. 有希望的，有前途的
aggregate	*v*. 聚集，集合，合计
ideally	*adv*. 理想地
robust	*adj*. 精力充沛的
turnover	*n*. 流通量，营业额

B. Useful expressions

label as	分类，标注
trot out	提出……供考虑
make sense to	有意义
thanks to	由于
across borders and oceans	遍布世界
have a good handle on	易于操控
evolve into	演化成
out of phase	异相地，不协调地
half the battle	成功一半

be tied in with	系在一起
distinguish from	区分
translate into	翻译成，转化为
set the standard for	定下标准
make up of	组成
as such	同样地，同量地

C. Technical terms and proper names

enterprise resource planning (ERP)	企业资源规划
enterprise system (ES)	企业系统
enterprise software (ES)	企业软件
manufacturing resource planning (MRP II)	制造资源计划
material requirements planning (MRP)	物料需求计划
seamless integration	无缝集成
return on investment (ROI)	投资回报率
islands of information	信息孤岛
bill of material (BOM)	物料清单
business process reengineering (BPR)	业务流程重组
closed-loop MRP	闭环 MRP
sales forecasting	销售预测
master production scheduling	主生产计划
accounts receivable	应收账款
sales and operation planning	销售和营运计划
rough-cut capacity planning (RCCP)	粗能力计划
accounts payable	应付账款
advanced planning system	远景规划系统
general leger	总分类账
supplier rating system	供应商评估系统
capacity requirements planning	能力需求计划
customer relation management (CRM)	客户关系管理

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. ERP began life in the 1960s as ________, an outgrowth of early efforts in bill of material processing.

2. Enterprise Resource Planning (ERP) predicts and balances demand and ________.

3. The goals of ERP include high levels of customer service, productivity, cost reduction, and __________.

4. Closed-loop MRP has provisions for __________ from the execution functions back to the planning functions. Plans can then be altered when necessary, thereby keeping priorities valid as conditions change.

5. Enterprise Resource Planning (ERP) is an enterprise-wide set of forecasting, planning, and scheduling tools, which: Links customers and suppliers into a complete ________, employs proven processes for ________ and coordinates sales marketing, operations, logistics, ________, finance, product development, and human resources.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. Manufacturing Resource Planning (MRP Ⅱ) predicts and balances demand and supply. ( )

2. Material Requirements Planning (MRP) involves three additional elements: sales and operations planning, financial interface, simulation. ( )

3. The goals of MRP Ⅱ include high levels of customer service, productivity, cost reduction, and inventory turnover, and it provides the foundation for effective supply chain management and e-commerce. ( )

4. The second step of ERP is Closed-loop MRP. ( )

5. Enterprise Resource Planning (ERP) predicts and balances demand and supply. It is an enterprise-wide set of forecasting, planning, and scheduling tools. ( )

Ⅲ. Match each of the following terms with the appropriate definition.

ERP	MRP	losed-Loop MRP	MRP Ⅱ

1. ______ Techniques for helping plan capacity requirements were tied in with Material Requirements Planning.

2. ______ An outgrowth of early efforts in bill of material processing.

3. ______ A method for the effective planning of all resources of a manufacturing company.

4. ______ An enterprise-wide set of management tools that balances demand and supply.

Ⅳ. Answer the following questions.

1. Describe the difference and relationship of ERP and ES.

2. Explain the evolution of ERP.

3. Compare and contrast MRP with MRP Ⅱ.

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

outgrowth	emergent	promising
evolutionary	definitive	robust
watershed	profound	regarding

1. The overwhelming feeling is just deep, ________ shock and anger.

2. The ________ countries of Africa are now becoming politically independent and modernized.

3. No one has come up with a ________ answer as to why this should be so.

4. Tonight could prove to be a ________ for the international career of Barnes.

5. The manufacture of this material is an ________ of the space industry.

6. I wrote a letter ________ my daughter's school examinations.

7. The school honored one of its brightest and most ________ former pupils yesterday morning.

8. We've always specialized in making very _______, simply designed machinery.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "demand" and fill in the sentences with the right word.

Example: When our customer **demand changes**, we and our suppliers, can manage changes to our schedules on a very coordinated and controlled basis.

1. Mr Byers last night demanded an immediate ex______ from the Education Secretary.

2. He said the task of reconstruction would demand much pa______, hard work and sacrifice.

3. There would be fewer international crises demanding his at______.

4. All parent-child relationships demand excellent communication sk______.

5. The developing nations of Africa and Latin America demand the na______ of certain industries.

6. The tendency of victims of crime to organize and demand stricter pu_____ has been a growing one in the United States.

Summary of the Text

Ⅶ. Choose the best one of the four Choices given to fill in each blank.

Enterprise resource planning (ERP) systems attempt to integrate several data sources and processes of an organization __1__ a unified system. A __2__ ERP system will use multiple components of computer software and hardware to achieve the integration. A key ingredient of most ERP systems is the use of __3__ unified database to store data for the various system

modules.

The two key components of an ERP system __4__ a common database and a modular software design. A common database is the system that allows __5__ department of a company to store and __6__ information in real-time. Using a common database allows information to be more __7__ , accessible, and easily shared. Moreover, a modular software design is a __8__ of programs that can be added on an individual __9__ to improve the efficiency of the business. This improves the business by adding functionality, mixing and matching programs from different vendors, and __10__ the company to choose which modules to implement. These modular software designs link into the common database, so that all of the information between the departments is accessible in real time.

1. A. into	B. with	C. in	D. on
2. A. common	B. special	C. particular	D. typical
3. A. an	B. a	C. the	D. some
4. A. is	B. was	C. were	D. are
5. A. every	B. many	C. some	D. all
6. A. take	B. put	C. retrieve	D. pick
7. A. rely	B. relied	C. reliable	D. relying
8. A. variety	B. kind	C. sort	D. series
9. A. base	B. basis	C. bases	D. based
10. A. allows	B. allow	C. allowing	D. allowed

Translation

Ⅷ. Translate the following into Chinese.

1. There's a lot of sloppy terminology flying around today in the business press, and one misnomer is to label enterprise-wide transaction processing software systems as ERP.

2. It is a strong contributor to America's amazing economic performance of the 1990s and the emergence of the New Economy.

3. When our customer demand changes, we and our suppliers can manage changes to our schedules on a very coordinated and controlled basis.

4. This is important, because in a manufacturing enterprise, change is not simply a possibility or even a probability.

5. Its goals include high levels of customer service, productivity, cost reduction, and inventory turnover, and it provides the foundation for effective supply chain management and e-commerce.

Chapter 17
Supply Chain Management

Pre-reading Questions

1. What is a supply chain?
2. Describe the model of supply chain (pull and push).
3. What are the objectivities of a supply chain?

17.1 What Is a Supply Chain

A supply chain consists of all states involved, directly or indirectly, in fulfilling a customer request. The supply chain not only includes the manufacturer and suppliers, but also transporters, warehouses, retailers, and customers themselves. Within each organization, such as a manufacturer, the supply chain includes all functions involved in filling a customer request. These functions include, but are not limited to, new product development, marketing, operations, distribution, finance, and customer service.

Consider a customer walking into a Wal-Mart store to purchase detergent. The supply chain begins with the customer and his or her need for detergent. The next stage of this supply chain is the Wal-Mart retail store that the customer visits. Wal-Mart stocks its shelves using inventory that may have been supplied from a finished-goods warehouse that Wal-Mart manages or from a distributor using trucks supplied by a third party. The distributor in turn is stocked by the manufacturer (say, Procter & Gamble [P & G] in this case).The P & G manufacturing plant receives raw material from a variety of suppliers, who may themselves have been supplied by lower-tier suppliers. For example, packaging material may come from Tenneco, and Tenneco might receive raw material to manufacture the packaging from other suppliers.

A supply chain is dynamic and involves the constant flow of information, product, and funds between different stages. Each stage of the supply chain performs different processes and interacts with other stages of the supply chain. Wal-Mart provides the product, as well as pricing and availability information, to the customer. The customer transfers funds to Wal-Mart. Wal-Mart conveys point-of-sale (POS) data as well as replenishment orders to the distribution center (DC), which transfers the replenishment order via trucks back to the store. Wal-Mart transfers funds to the distributor after the replenishment. The distributor also provides pricing information and sends delivery schedules to Wal-Mart. Similar information, material, and fund flows take place across the entire supply chain.

In another example, when a customer purchases on-line from Dell Computer, the supply chain

includes, among others, the customer, the Web page that takes the customer's order, the Dell assembly plant, and all of Dell's suppliers and their suppliers. The Web page provides the customer with information regarding pricing, product variety, and product availability. Having made a product choice, the customer enters the order information and pays for the product. The customer may later return to the Web page to check the status of the order. Stages further up the supply chain use customer order information to fill the order. That process involves an additional flow of information, in the process generating profits for itself. Supply chain activities begin with a customer order and end when a satisfied customer has paid for his or her purchase. The term supply chain conjures up images of product, or supply, moving from suppliers to manufacturers to distributors to retailers to customers along a chain. It is important to visualize information, funds, and product flows along both directions of his chain. The term may also imply that only one player is involved at each stage. In reality, a manufacturer may receive material from several suppliers and then supply several distributors. Therefore, most supply chains are actually networks. It may be more accurate to use the terms supply network or supply web to describe the structure of most supply chains.

A typical supply chain may involve a variety of stages. These supply chain stages include the following:

- Customers
- Retailers
- Wholesalers/distributors
- Manufacturers
- Component/raw material suppliers

Each stage need not be present in a supply chain. The appropriate design of the supply chain will depend on both the customer's needs and the roles of the stages involved in filling those needs. In some cases, such as Dell, a manufacturer may fill customer orders directly. Dell builds to order, that is, a customer order initiates manufacturing at Dell. Dell does not have a retailer, wholesaler, or distributor in its supply chain. In other cases, such as the mail order company L.L. Bean, manufacturers do not respond to customer orders directly. L.L. Bean maintains an inventory of products from which it fills customer orders. Compared with the Dell supply chain, the L.L. Bean supply chain contains an extra stage (the retailer, L.L. Bean itself) between the customer and the manufacturer. In the case of a small retail store, the supply chain may also contain a wholesaler or distributor between the store and the manufacturer.

17.2 The Objectives of a Supply Chain

The objective of every supply chain is to maximize the overall value generated. The value a supply chain generates is the difference between what the final product is worth to the customer and the effort the supply chain expends in filling the customer's request. For most commercial

supply chains, value will be strongly correlated with supply chain profitability,

The difference between the revenue generated from the customer and the overall cost across the supply chain. For example, a customer purchasing a computer from Dell pays $2000, which represents the revenue the supply chain receives. Dell and other stages of the supply chain incur costs to convey information, produce components, store them, transport them, transfer funds, and so on. The difference between the $2000 that the customer paid and the sum of all costs incurred by the supply chain to produce and distribute the computer represents the supply chain profitability. Supply chain profitability is the total profit to be shared across all supply chain stages. The higher the supply chain profitability, the more successful the supply chain. Supply chain success should be measured in terms of supply chain profitability and not in terms of the profits at an individual stage.

We define the success of a supply chain in terms of supply chain profitability. The next logical step is to look for sources of revenue and cost. For any supply chain, there is only one source of revenue: the customer. The customer is the only real point of positive cash flow in a supply chain. In the Wal-Mart example, the customer purchasing detergent is the only one providing positive cash flow for the supply chain, given that different stages have different owners. When Wal-Mart pays its supplier, it is taking a portion of the funds the customer provides and passing that money on to the supplier. This cash transfer adds to the supply chain's costs. All flows of information, product, or funds generate costs within the supply chain. Therefore, the appropriate management of there flows is a key to supply chain success. Supply chain management involves the management of flows between and among stages in a supply chain to maximize total profitability.

17.3　Decision Phases in a Supply Chain

Successful supply chain management requires several decisions relating to the flow of information, product, and funds. These decisions fall into three categories or phases, depending on the frequency of each decision and the time frame over which a decision phase has an impact.

1. Supply Chain Strategy or Decision

During this phase, a company decides how to structure the supply chain. It decides what the chain's configuration will be and what processes each stage will perform. Decisions made during this phase are also referred to as strategic supply chain decisions. Strategic decisions made by companies include the location and capacities of production and warehousing facilities, products to be manufactured or stored at various locations, modes of transportation to be made available along different shipping legs, and type of information system to be made available along different shipping legs, and type of information system to be utilized. A firm must ensure that the supply chain configuration supports its strategic objectives during this phase. Dell's decisions regarding the location and capacity of its manufacturing facilities, warehouses, and supply sources are all supply chain design or strategic decisions. Supply chain design decisions are typically made for

the long term (a matter of years) and are very expensive to alter on short notice. Consequently, when companies make these decisions, they must take into account uncertainty in anticipated market conditions over the next few years.

2. Supply Chain Planning

As a result of the planning phase, companies define a set of operating policies that govern short-term operations. For decisions made during this phase, the supply chain's configuration determined in the strategic phase is fixed. This configuration establishes constraints within which planning must be done. Companies start the planning phase with a forecast for the coming year (or a comparable time frame) of demand in different markets. Planning includes decisions regarding which markets will be supplied from which locations, the planned buildup of inventories, the subcontracting of manufacturing, the replenishment and inventory policies to be followed, the policies that will be enacted regarding backup locations in case of a stockout, and the timing and size of marketing promotions. Dell's decisions regarding the markets a given production facility will supply and target production quantities at different locations are classified as planning decisions. Planning establishes parameters within which a supply chain will function over a specified period of time. In the planning phase, companies must include uncertainty in demand, exchange rates, and competition over this time horizon in their decisions. Given a shorter time horizon and better forecasts than the design phase, companies in the planning phase try to incorporate whatever flexibility may have been built into the supply chain in the design phase and exploit it to optimize performance in the shorter term.

3. Supply Chain Operation

The time horizon here is weekly or daily, and during this phase companies make decisions regarding individual customer orders. At the operational level, supply chain configuration is considered fixed and planning policies already defined. The goal of supply chain operations is to implement the operating policies in the best possible manner. During this phase, firms allocate individual orders to inventory or production, set a date that an order is to be filled, generate pick lists at a warehouse, allocate an order to a particular shipping mode and shipment, set delivery schedules of trucks, and place replenishment orders. Because operational decisions are being made in the short term (minutes, hours, or days), there is often less uncertainty about demand information. The goal during the operation phase is to exploit the reduction of uncertainty and optimize performance within the constraints established by the configuration and planning policies.

The design, planning, and operation of a supply chain have a strong impact on overall profitability and success. Consider Dell Computer. In 1993, Dell performed poorly, and its stock price dropped sharply. This led Dell management to focus on improving the design, planning, and operation of the supply chain, resulting in significantly improved performance. Both profitability and the stock price subsequently have soared due to this increase in performance.

17.4 The Importance of Supply Chain Flows

There is a close connection between the design and management of supply chain flows (product, information, and cash) and the success of a supply chain. Dell Computer is an example of a firm that has successfully used good supply chain practices to support its competitive strategy. In contrast, Quaker Oats is an example in which the inability to design and manage flows appropriately in the supply chain led to the failure of its acquisition of Snapple.

Dell was established in 1984. By 1998, it had grown into a $12 billion company. Since 1993, Dell has experienced earnings growth of more than 30 percent per year over the next five years. Dell's stock price has also grown significantly since 1993. Dell has attributed a significant part of its success to the way it manages flows — product, information, and cash — within its supply chain.

Dell's basic supply chain model is direct sales to customer. As distributors and retailers are bypassed, the Dell supply chain has only three stages — customer, manufacturer, and suppliers.

Because Dell is in direct contact with its customers, it has been able to finely segment them and analyze the need and profitability of each segment. Close contact with its customers and an understanding of customers' need also allows Dell to develop better forecasts. To further improve the match between supply and demand, Dell makes an active effort to steer customers in real time, on the phone or via the Internet, toward PC configurations that can be built given the components available.

On the operational side, inventory turns is a key performance measure that Dell watches very closely. Each computer chip carries a data code to indicate how old a part is. Dell carries only about 10 days' worth of inventory; in contrast, the competition, selling through retailers, has been carrying in the vicinity of 80 to 100 days. If Intel introduces a new chip, the low level of inventory allows Dell to go to market with a PC containing the chip faster than the competition. If prices suddenly drop, as they did in the early part of 1998, Dell has less inventory that loses value relative to its competitors. For some products, such as monitors manufactured by Sony, Dell maintains no inventory. The transportation company simply picks up the appropriate number of computers from Dell's Austin plant and monitors from Sony's factory in Mexico, matches them by customer order, and delivers them to the customer. This procedure allows Dell to save time and money associated with the extra handling of monitors.

The success of the Dell supply chain is facilitated by sophisticated information exchange. Dell provides real-time data to suppliers on the current state of demand. Suppliers are able to access their components' inventory levels at the factories along with daily production requirements. Dell has created customized Web pages so that its major suppliers can view demand forecasts and other customer-sensitive information, thus helping suppliers to get a better idea of customer demand and better match their production schedules to that of Dell. The company has production concentrated in five manufacturing centers: Austin, Texas; Brazil; China; Ireland; and Malaysia. Because demand at each location is relatively large and stable, allowing Dell to maintain low levels of component

inventories. In some cases, Dell carries only hours of component inventory at its factory.

Dell low levels of inventory also help ensure that defects are not introduced into a large quantity of product. When a new product is launched, supplier engineers are stationed right in the plant. If a customer calls in with a problem, production is stopped and design flaws are fixed in real time. As there is no finished product in inventory, the amount of defective merchandise produced is minimized.

Dell's also outsources service and support to third-party providers. To ensure a high quality of service, Dell coordinates the delivery of the parts the customer requires with the arrival of the service person. Once again, a coordinated flow of information and material minimizes the cost necessary to provide a high level of service.

Clearly, Dell's supply chain design and appropriate management of product, information, and cash flows play a key role in the company's success. This approach has positioned Dell very well within the PC industry. Given that good product performance is common in the PC industry, the competitive battlefield is now focused on service delivery and supply chain efficiency.

Quaker Oats, with its acquisition of Snapple, provides an example in which failure to design and manage supply chain flows led to financial failure. Quaker owns Gatorade, the top-selling brand in the sports drink segment. In December 1994, Quaker purchased Snapple at a cost of $1.7 billion. Snapple sold all-natural drinks. At the time, Gatorade was very strong in the south and the southwest of the United States, while Snapple was strong in the northeast and on the west coast.

Quaker announced that it hoped to exploit synergies between the two distribution systems to gain efficiencies, but the company was unable to remedy several problems that prevented it from achieving these synergies. Gatorade was manufactured in plants owned by Quaker, while Snapple was produced under contact by outside plants. Gatorade sold significant amounts through supermarkets and grocery stores, while Snapple sold primarily through restaurants and independent retailers. Over the two years following its acquisition of Snapple, Quaker was unable to gain much synergy between the two distribution systems. In trying to merge two disparate distribution systems, Quaker helped neither and may have hurt both. Just 28 months later, Quaker sold Snapple to Triarc Companies for about $300 million, about 20 percent of the purchase price. The inability to achieve synergies between the two supply chains was a significant reason for the failure of Snapple at Quaker.

Technical Notes to the Text

1. supply chain，供应链。供应链是为了实现一定的目标(如满足消费者的需求)而采取的一系列相互依赖的步骤的集合。

2. distribution center (DC)，配送中心。配送中心是指接受供应者所提供的多品种、大批量的货物，通过储存、保管、分拣、配货，以及流通加工、信息处理等作业后，将按订货要求配齐的货物送交顾客的组织机构和物流设施。配送中心在以下几个方面发挥较好的作用：

减少交易次数和流通环节；产生规模效益；减少客户库存，提高库存保证程度；与多家厂商建立业务合作关系，能有效而迅速地反馈信息，控制商品质量。配送中心是现代电子商务活动中开展配送活动的物质技术基础。

3. supply chain management (SCM)，供应链管理。供应链管理是产生订单、订单执行、订单完成、产品、服务或信息分发过程的合作，供应链内的相互依赖创造了一个“扩展的企业”。原材料供应商、流通渠道伙伴(批发商、分销商、零售商)及消费者本身都是供应链管理的主要角色。

4. point-of-sale (POS)，销售点终端。POS 是一种多功能终端，把它安装在信用卡的特约商户和受理网点中与计算机联成网络，就能实现电子资金自动转账。它具有支持消费、预授权、余额查询和转账等功能，使用起来安全、快捷、可靠。POS 主要有以下几种类型：①消费 POS，具有消费、预授权、查询支付名单等功能，主要用于特约商户受理牡丹卡消费；②转账 POS，具有财务转账和银行卡转账等功能，主要用于单位财务部门。

Word Bank to the Text

A. Useful new words

retailer	*n*. 零售商
detergent	*n*. 清洁剂，去垢剂
dynamic	*adj*. 动态的
availability	*n*. 可用性，有效性
replenishment	*n*. 补给，补充
visualize	*v*. 形象，形象化，想象
initiate	*v*. 开始，发动
wholesaler	*n*. 批发商
revenue	*n*. 收入，税收
incur	*v*. 招致
uncertainty	*n*. 无常，不确定，不可靠
constraint	*n*. 约束，强制
flexibility	*n*. 弹性，适应性，机动性
optimize	*v*. 使最优化
soar	*v*. 翱翔，骤升
inability	*n*. 无能，无力
acquisition	*n*. 获得，获得物
bypass	*n*. 忽视，不留意
vicinity	*n*. 邻近，附近
defective	*adj*. 有缺陷的
launch	*v*. 开办，发动
merchandise	*n*. 商品，货物

synergy	*n*. 协同，配合
remedy	*v*. 治疗，补救

B. Useful expressions

fulfill a request/need	达到要求
involve in	包括
the flow of information	信息流动
convey to	传递
fill the order	满足订单
conjure up	想象，推想
in reality	实际上，事实上
worth the effort	值得付出
correlate with	把……与……联系起来
add to	增加
have an impact on	对……起作用，产生效果
on short notice	忽然，急忙
take into account	考虑到
be classified as	被归类
result in	导致
increase in performance	性能的提升
be attributed to	归因于
in direct contact with	与……有直接关系
in the vicinity of	在邻近
associate with	联合
coordinate with	协调
with the arrival of	随着……的到来

C. Technical terms and proper names

supply chain	供应链
distribution center (DC)	配送中心
supply chain management (SCM)	供应链管理
point-of-sale (POS)	销售点终端

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. The success of the Dell supply chain is facilitated by sophisticated ________. Dell provides

data to suppliers on the current state of demand.

2. A typical supply chain may involve a variety of stages, include the following: ________ , retailers, ________ , manufacturers and ________ .

3. The objective of every supply chain is to ________ the overall value generated.

4. Dell's basic supply chain model is direct sales to ________. As distributors and retailers are bypassed, the Dell supply chain has only three stages — customer,________, and suppliers.

5. Successful supply chain management requires several decisions relating to the flow of ________, product, and ________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. A supply chain is static and involves the constant flow of information, product, and funds between different stages. (　　)

2. Successful supply chain management requires several decisions relating to the flow of information, product, and funds. These decisions fall into four categories or phases. (　　)

3. Because Dell is in indirect contact with its customers, it has been able to finely segment them and analyze the need and profitability of each segment. (　　)

4. LM involves addressing the product strategy, product development, supply chain, manufacturing and product distributions for the production of diverse product. (　　)

5. The failure reason of Quaker Oats is the inability to design and manage flows appropriately in the supply chain. (　　)

Ⅲ. Match each of the following terms with the appropriate definition.

supply chain	supply chain strategy
supply chain planning	supply chain operation

1. _______ Consists of all states involved in fulfilling a customer request.

2. _______ Companies make decisions regarding individual customer orders.

3. _______ How to structure the supply chain, what the chain's configuration will be and what processes each stage will perform.

4. _______ Companies define a set of operating policies that govern short-term operations.

Ⅳ. Answer the following questions.

1. Describe the difference between "pull" supply chain and "push" supply chain.

2. Describe the difference of supply chain between Wal-Mart and Dell.

3. Analysis the success factors of Dell and the failure factors of Quaker Oats.

Vocabulary

V. Fill in the blanks with the words given below. Change the form where necessary.

incur	detergent	bypass
optimize	initiate	defective
revenue	vicinity	inability

1. There were a hundred or so hotels in the _________ of the railway station.
2. The trip was ________ by the manager of the community centre.
3. One study said the government would gain about $12 billion in tax ________ over five years.
4. She falls in love and ________ the anger of her father.
5. The new systems have been ________ for running Microsoft Windows.
6. Her ________ to concentrate could cause an accident.
7. A growing number of employers are trying to _______ the unions altogether.
8. A report has pointed out the ________ of the present operating system.

Collocation

VI. Look at the following sentences taken from the text. Just think about what else you can "initiate" and fill in the sentences with the right word.

Example: An **interruption** of a program that is **initiated** by a software instruction.

1. Louis Cheskin published the groundbreaking Color For Profit in 1951 and initiated a scientific ap______ to color and design.
2. In 1939, Einstein wrote his famous letter to US President Roosevelt in which he urged him to initiate an American nuclear research pr______.
3. However, the commons had no power to initiate le______. It could only ask the monarch to grant requests.
4. There is no doubt that people have instinctive responses to colors and shapes. But just how these responses initiate the pu______ of a box of pudding mix or a jar of skin cream is not well understood.
5. The By 1967, National Councillor Willy Bretscher tried to initiate a ha______ with the UN and possible membership.
6. We think it wrong on fundamental ethical principles to initiate the de______ of such a weapon.

Summary of the Text

VII. Choose the best one of the four answers given to fill in each blank.

Supply chain management (SCM) is the process of planning, implementing and controlling

the operations of the supply chain as efficiently as __1__ SCM spans all movement and storage of __2__ materials, work-in-process inventory, and finished goods from point-of-origin to point-of-consumption.

Traditionally, companies in a supply network concentrate __3__ the inputs and outputs of the processes, with __4__ concern for the internal management working of other individual players. Therefore, the choice of an __5__ management control structure is known to __6__ local firm performance. During the past decades, globalization, outsourcing and information technology have __7__ many organizations, such as Dell and Hewlett Packard, to successfully operate solid __8__ supply networks in which each specialized business partner focuses on only a few key strategic activities. This inter-organizational supply network can be acknowledged __9__ a new form of organization, many of __10__ increasingly find that they must rely on effective supply chains, or networks, to successfully compete in the global market and networked economy.

1. A. soon	B. possible	C. can	D. well
2. A. fine	B. dry	C. raw	D. good
3. A. on	B. in	C. over	D. with
4. A. much	B. more	C. few	D. little
5. A. efficiency	B. storage	C. internal	D. external
6. A. improve	B. help	C. impact	D. regard
7. A. able	B. enable	C. enabling	D. enabled
8. A. collaborative	B. collaboration	C. mutual	D. assistant
9. A. as	B. to	C. by	D. with
10. A. that	B. which	C. whose	D. whom

Translation

Ⅷ. Translate the following into Chinese.

1. The objective of every supply chain is to maximize the overall value generated.

2. The value a supply chain generates is the difference between what the final product is worth to the customer and the effort the supply chain expends in filling the customer's request.

3. Successful supply chain management requires several decisions relating to the flow of information, product, and funds.

4. For decisions made during this phase, the supply chain's configuration determined in the strategic phase is fixed.

5. Given that good product performance is common in the PC industry, the competitive battlefield is now focused on service delivery and supply chain efficiency.

Chapter 18
Customer Relationship Management

Pre-reading Questions

1. What is customer relationship management? Why are customer relationships so important today?

2. Describe the advantages of customer relationship management.

3. Name four types of customers.

4. How many capabilities can a typical CRM software provide? Name each of them.

Businesses have always valued their customers, but today there is much greater appreciation of the importance of customers for the profitability of the enterprise. Because competitive advantage based on an innovative new product or service can be very short lived, companies are realizing that their only enduring competitive strength may be their relationships with their customers. Some say that the basis of competition has switched from who sells the most products and services to who "owns" the customer, and that customer relationships represent the firm's most valuable asset.

18.1 Customer Relationship Management Concepts

Many firms are turning to customer relationship management (CRM) to maximize the benefits of their customer assets. Customer relationship management is both a business and technology discipline for managing customer relationships to optimize revenue, profitability, customer satisfaction, and customer retention.

Simply defined, it is the process of acquiring, retaining and growing profitable customers. It requires a clear focus on the service attributes that represent value to the customer and that create loyalty.

Customer relationship management has several advantages over traditional mass-media marketing. It:

- reduces advertising costs.
- makes it easier to target specific customers by focusing on their needs.
- makes it easier to track the effectiveness of a given campaign.
- allows organizations to compete for customers based on service, not prices.
- prevents overspending on low-value clients or understanding on high-value ones.
- speeds the time it takes to develop and market a product (the marketing cycle).

- improves use of the customer channel, thus making the most of each contact with a customer.

18.2 Four Types of Customers

CRM allows a company to address all of the types of customers it serves at different points in their life cycle and to choose the marketing program that best fits a customer’s attitude toward the company and willingness to purchase its products and services. Four types of CRM programs enable the company to win back customers who have defected or are planning to, to create loyalty among existing customers, to up-sell or cross-sell services to these customers and to prospect for new customers.

18.2.1 Win Back or Save

This is the process of convincing a customer to stay with the organization at the point they are discontinuing service or convincing them to rejoin once they have left. Of the four categories of campaigns, win back is the most time-sensitive. Research indicates that a win-back campaign is four times likely to succeed if contact is made within the first week following a defection than if it is made in the fourth week.

Selectivity is the other essential characteristic of a successful win-back campaign. Leading organizations often filter their prospects for contact to exclude customers who have frequently switched (churners), who have bad credit ratings or whose usage is low.

Most new selection techniques are allowing organizations to trim back their contact lists, but one trend is extending them. In the past years, organizations would ignore customers who had a significant decline in usage or who had discontinued some services as long as they remained customers. The organizations assumed that such customers were merely switching to another product. Without good data and the ability to analyze it, there was no way to disprove this assumption. Recent work in this area has shown that many of these consumers are either reducing overall usage or, worse, migrating to a competitor’s product. To preserve the revenue stream and prevent the customer from becoming a “traditional” win-back candidate, a few organizations are now including partial disconnects and reduced-usage customers in their win-back campaigns.

18.2.2 Prospecting

Prospecting is the effort to win new, first-time customers. Apart from the offer itself, the three most critical elements of a prospecting campaign are segmentation, selectivity and sources. It is essential to develop an effective needs-based segmentation model that allows the organization to effectively target the offer. Without this focused approach, the organization either fails to achieve an adequate acceptance or rate on the offer or spends too much on promotions, advertising and concessionary pricing. It is advisable to achieve a 95 percent confidence rate before embarking on

a prospecting effort.

Selectivity is as important to prospecting as it is to win back. Needs-based segmentation defines what the customer wants from the organization and profit-based segmentation, defines how valuable the customer is helps the organization decide how much it is willing to spend to get that customer. Pre-scoring a consumer credit rating is one of the techniques that organizations can use to determine the latter. Organizations have traditionally experienced rates of uncollectibles ranging from 4 to 8 percent. To reduce this figure, they are following an example set by the retail industry, which developed credit-scoring algorithms to apply to prospective consumers. Retail companies that do this have cut their uncollectibles in half — from 6 to 3 percent — on average. Today's organizations hope to meet with similar success.

18.2.3 Loyalty

Loyalty is the category in which it is most difficult to gain accurate measures. The organization is trying to prevent customers from leaving and uses three essential elements: value-based and need-based segmentation and predictive churn models. Value-based segmentation allows the organization to determine how much it is willing to invest in retaining a customer's loyalty. It is possible that the organization will invest nothing in those customers, so it deems to be marginally profitable and will actively encourage unprofitable customers to leave.

Once the customer has passed the value-based segmentation screening, the organization can use needs-based segmentation to offer a customized loyalty program. Affinity programs, such as airline miles and hotel points, are some of the most popular. Besides affinity programs, organizations will often offer customized billing, special help lines or back-end loaded credits as a means of encouraging loyalty. It is important to note that most of these offers are based more on a customer's revenue level than tailored to their segments. However, as organizations focus more on the needs of individual customers, they find that they are able to achieve the same level of loyalty with less investment.

The final component of a successful loyalty campaign is the development of a predictive churn model. Using the vast amount of demographic data and usage history available for the existing base of customers, it is possible to predict customer attrition. Through the use of advanced data-mining tools, organizations can develop models that identify vulnerable customers, which can then be targeted for a loyalty campaign or offered alternative products. Organizations should achieve a confidence ratio of 70 percent or greater with their churn models before implementing campaigns. At levels lower than this the cost far outweighs the potential increase in gross profit.

18.2.4 Cross-Sell/Up-Sell

This CRM program is also known as increasing wallet share or the amount the customer spends with you. The purpose is to identify complementary offerings that a customer would like. For instance, a basic long-distance customer could be a candidate to buy Internet access. The

nature of the offer is determined by the customer's needs-based segment, usage pattern and reaction to previous contacts. Once the composition of the offer is determined and the contact medium is agreed to, then the organization directly presents that offer to the customer. Up-selling is similar, but instead offering a complementary product, the organization offers an enhanced one. For example, replacing an analog data line with ISDN is a good example of an up-sell.

Cross-sell/up-sell campaigns are important because the customers targeted already have a relationship with the organization. They are less likely to see the offer as a commodity and are thus more willing to pay a premium for it. In financial terms, when a customer accepts a cross-sell or up-sell offer, that customer begins to become much more profitable. At the outset of this relationship, the customer reduces gross margin by 3 percent (based on acquisition costs); within three years, this customer is enhancing gross margin by 7 percent.

18.3 Customer Relationship Management (CRM) Software

Commercial customer relationship management (CRM) software packages range from niche tools that perform limited functions, such as personalizing Web sites for specific customers, to large-scale enterprise applications that capture myriad interactions with customers, analyze them with sophisticated reporting tools, and link to other major enterprise applications, such as supply chain management and enterprise systems. The more comprehensive CRM packages contain modules for partner relationship management (PRM) and employee relationship management (ERM).

Customer relationship management (CRM) systems typically provide software and online tools for sales, customer service, and marketing. Their capabilities include the following.

18.3.1 Sales Force Automation (SFA)

Sales force automation modules in CRM systems help sales staff increase their productivity by focusing sales efforts on the most profitable customers, those who are good candidates for sales and services. CRM systems provide sales prospect and contact information, product information, product configuration capabilities, and sales quote generation capabilities. Such software can assemble information about a particular customer's past purchases to help the salesperson make personalized recommendations. CRM software enables customer and prospect information to be shared easily among sales, marketing, and delivery departments. It increases each salesperson's efficiency in reducing the cost per sale as well as the cost of acquiring new customers and retaining old ones. CRM software also has capabilities for sales forecasting, territory management, and team selling.

18.3.2 Customer Service

Customer service modules in CRM systems provide information and tools to make call centers, help desks, and customer support staff more efficient. They have capabilities for assigning and managing customer service request.

One such capability is an appointment or advice telephone line: when a customer calls a

standard phone number, the system routes the call to the correct service person, who inputs information about that customer into the system only once. Once the customer's data are in the system, any service representative can handle the customer relationship. Improved access to consistent and accurate customer information helps call centers handle more calls per day and decreases the duration of each call. Thus, call centers and customer service groups can achieve greater productivity, reduce transaction time, and higher quality of service at lower cost. The customer is happier because he or she spends less time on the phone restating his or her problem to customer service.

CRM systems may also include Web-based self-service capabilities: the company Web site can be set up to provide inquiring customers personalized support information as well as the option to contact customer service staff by phone for additional assistance.

18.3.3 Marketing

Customer relationship management (CRM) systems support direct-marketing campaigns by providing capabilities for capturing prospect and customer data, for providing product and service information, for qualifying leads for targeted marketing, and for scheduling and tracking direct-marketing mailings or E-mail. Marketing modules would also include tools for analyzing marketing and customer data — identifying profitable and unprofitable customers, designing products and services to satisfy specific customer needs and interests, and identifying opportunities for cross-selling, up-selling, and bundling.

Cross-selling is the marketing of complementary products to customers. Up-selling is the marketing if higher-value products or services to new or existing customers. Bundling is one kind of cross-selling in which a combination of products is sold as a bundle at a price lower than the total cost of the individual products.

18.4 Operational and Analytical CRM

All of the applications of CRM support either operational or analytical aspects of customer relationship management. Operational CRM includes customer-facing applications such as tools for sales force automation, call center and customer service support, and marketing automation. Analytical CRM includes applications that analyze customer data generated by operational CRM applications to provide information for improving business performance management. Table 18-1 provides more specific examples of operational and analytical CRM functions.

Table 18-1 Examples of Operational Versus Analytical CRM

Operational CRM	Analytical CRM
campaign management	develop customer segmentation strategies
e-marketing	develop customer profits
account and contact management	analyze customer profitability

(续表)

Operational CRM	Analytical CRM
lead management	analyze product profitability
telemarketing	identify cross-selling and up-selling opportunities
teleselling	select the best marketing, service, and sales channels for each customer group
e-selling	identify trends in sales cycle length, win rate, and average deal size
field service dispatch	analyze service resolution times, service levels based on communication channel, and service activity by product line and account
customer care and help desk	analyze leads generated and conversion rate
contract management	identify chum problems

Technical Notes to the Text

1. customer relationship management (CRM)，客户关系管理。CRM 首先是一种管理理念，其次才是一种技术。首先，从企业经营管理的角度来看，CRM 是企业的一项商业策略。CRM 按照客户的分割情况有效地组织企业资源，培养以客户为中心的经营行为及实施以客户为中心的业务流程，并以此为手段来提高企业的获利能力、收入及客户的满意度。其次，从技术的角度来看，CRM 是企业在营销、销售和服务业务范围内，对现实的和潜在的客户关系及业务伙伴关系进行多渠道管理的一系列过程和技术。

2. win back campaigns，赢回顾客。

3. customer segmentation，客户细分。客户细分是指根据客户属性划分的客户集合。它既是客户关系管理的重要理论组成部分，又是其重要的管理工具。它是分门别类研究客户、进行有效客户评估、合理分配服务资源、成功实施客户策略的基本原则之一，为企业充分获取客户价值提供理论和方法指导。顾客细分理论的原理是：每类产品的顾客群不是一个群体，根据顾客群的文化观念、消费收入、消费习俗、生活方式的不同细分新的类别，企业根据消费者制定品牌推广战略和营销策略，将资源针对目标顾客集中使用。客户细分包括：①needs-based segmentation，基于需求的客户细分；②profit-based segmentation，基于利益的客户细分；③value-based segmentation，基于价值的客户细分等几种方法。

4. loyalty，客户忠诚度。客户忠诚度可以说是客户与企业保持关系的紧密程度，以及客户抗拒竞争对手吸引的程度。客户满意是客户对企业或其产品与服务的一种态度，而客户忠诚则是反映客户的行为。一般来说，忠诚的客户往往具有以下一些基本特征：周期性重复购买；同时使用多个产品和服务；向其他人推荐企业的产品；对于竞争对手的吸引视而不见；对企业有着良好的信任，能够在服务中容忍企业的一些偶尔失误。

5. cross-sell，交叉销售。所谓交叉销售，就是发现现有客户的多种需求，并通过满足其需求而实现销售多种相关的服务或产品的营销方式。促成交叉销售的各种策略和方法即为“交叉营销”。简而言之，交叉销售是说服现有的顾客去购买另一种产品，也是根据客人的多种需求，在满足其需求的基础上实现销售多种相关的服务或产品的营销方式。

6. up-sell，提升销售。提升销售是指电话销售人员通过对客户需求的深度挖掘和对客户工作环境的深度理解，向客户推荐产品价值更高的产品，更好地满足客户的需求，以求达到双赢。

7. partner relationship management (PRM)，伙伴关系管理。它是 CRM 系统的销售、客户服务及其他企业业务功能向合作伙伴的延伸，可促进更具有合作性的渠道伙伴关系。

8. employee relationship management (ERM)，员工关系管理。

9. bundling，绑定。绑定即把两个及以上商品或服务打包后，以一个价格进行销售。

10. sales force automation (SFA)，销售自动化。销售自动化指在所有的销售渠道中，以包括现场/移动销售(field/mobile sales)、内部销售/电话销售(inside sales/telesales)、销售伙伴(selling partner)、在线销售(web selling)和零售应用技术来达到提升销售的目的。

11. call center，呼叫中心。呼叫中心又叫作客户服务中心，它是一种基于CTI(computer telephony integration，计算机电话集成)技术，充分利用通信网和计算机网的多项功能集成，并与企业连为一体的一个完整的综合信息服务系统，利用现有的各种先进的通信手段，有效地为客户提供高质量、高效率、全方位的服务。呼叫中心通过信息共享，能快速、准确地满足用户查询和申报服务，使服务数量和服务质量都大大提高。而建立用户专属的服务档案和人性化的服务体系，能极大地提升客户满意度，从而促进用户忠诚度。此外，利用呼叫中心建立的庞大客户资料库，企业还可以进行电话、网络推销和市场调查，挖掘潜在用户。

12. operational CRM，操作型 CRM。操作型 CRM 系统使客户呼叫中心、客户服务部、市场营销部等部门的业务员在日常工作中能共享客户资源，减少信息流动滞留点，使呈现在客户印象中的企业为一个整体。其重点是创建一个客户数据库，这个数据库提供了客户与关系的一致性描绘并用专门的应用程序来提供这些信息，这些软件包括 SFA(销售自动化)和客户服务程序等。

13. analytical CRM，分析型 CRM。分析型 CRM 系统的用户不需要直接同客户打交道，而是从操作型 CRM 系统所产生的大量交易数据中提取有价值的各种信息，再利用数据挖掘技术建立各种行为预测模型，最后利用图表、曲线等对企业各种关键性能指标(key performance indicator，KPI)及客户市场分割情况向操作型应用发布，达到成功决策的目的。

Word Bank to the Text

A. Useful new words

appreciation	*n.* 正确评价，欣赏
innovative	*adj.* 创新的，革新(主义)的
retention	*n.* 保留，保持
loyalty	*n.* 忠诚，忠心
address	*v.* 向……致辞，演说，写姓名地址，从事，忙于
convince	*v.* 使确信，使信服
discontinue	*v.* 停止，中止

defection	*n*. 缺点，背信
selectivity	*n*. 选择性
filter	*v*. 过滤，筛选
trim	*v*. 整理，修整
migrate	*v*. 使移居，使移植
partial	*adj*. 部分的，偏袒的
disconnect	*v*. 拆开，分离，断开
prospect	*v*. 寻找，勘探
segmentation	*n*. 分割
concessionary	*adj*. 特许的，让步的
prescore	*v*. 先期评估
uncollectible	*adj*. 无法收集的，不可收回的
predictive	*adj*. 预言性的，成为前兆的
churn	*n*. 搅拌，搅动，激动感情
screen	*n*. 筛选
affinity	*n*. 密切关系，亲和力
demographic	*adj*. 人口统计学的
attrition	*n*. 摩擦，磨损，消耗
vulnerable	*adj*. 易受攻击的，易受……的攻击
complementary	*adj*. 补充的，补足的
premium	*n*. 奖赏，赠品
margin	*n*. 差数，(时间、金额等的)富余
enhance	*v*. 增加，加强
assemble	*v*. 集合，聚集
forecast	*v*. 预想，预测
consistent	*adj*. 一贯的，始终如一的
bundle	*n*. 捆，束，包　*v*. 捆扎
dispatch	*v*. 分派，派遣
chum	*n*. 密友，室友

B. Useful expressions

have advantages over	胜过，由于
compete for	为……竞争
contact with	接触，联系
stay with	继续下去，跟着……不离开
win back	恢复，夺得
embark on	从事，着手
make sth available for	使……可以享受某物，使……买得起某物
be the characteristic of	是……的特征

prevent from	阻止，妨碍
apart from	除……之外
set an example to	做……榜样，树立(好)榜样
on average	平均起来
deem to be	认为是
be determined by	由……决定
replace with	取代，以……代替
have a relationship with	与……有关系
pay a premium for	付佣金
at the outset of	在开始时

C. Technical terms and proper names

customer relationship management (CRM)	客户关系管理
win back campaigns	赢回活动
needs-based segmentation	基于需求的客户细分
profit-based segmentation	基于利润的客户细分
value-based segmentation	基于价值的客户细分
loyalty	忠诚度
cross-sell	交叉销售
up-sell	提升销售
partner relationship management (PRM)	伙伴关系管理
employee relationship management (ERM)	员工关系管理
bundling	捆绑销售
sales force automation (SFA)	销售自动化
call center	呼叫中心
operational CRM	操作型 CRM
analytical CRM	分析型 CRM

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. CRM is the process of acquiring, __________ and growing profitable customers. It requires a clear _________ on the service attributes that represent value to the customer and that create _________.

2. __________ is the effort to win new, first-time customers.

3. Win back or save is the process of convincing a customer to stay with the organization at

the point they are _________. Service or convincing them to rejoin once they have left. Of the four categories of campaigns, _________ is the most _________.

4. Through the use of advanced _________ tools, organizations can develop models that identify _________ customers, which can then be targeted for a _________ campaign or offered alternative products.

5. The final component of a successful loyalty _________ is the development of a predictive model.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. Loyalty is the effort to win new, first-time customers. (　　)

2. Up-selling is offering a complementary product to customer. (　　)

3. There are four types of customer. (　　)

4. Cross-sell is offering an enhanced product to customer. (　　)

5. Prospecting is the process of convincing a customer to stay with the organization at the point they are discontinuing service or convincing them to rejoin once they have left. (　　)

Ⅲ. Match each of the following terms with the appropriate definition.

up-selling	CRM
prospecting	win back

1. ______ the process of acquiring, retaining and growing profitable customers.

2. ______ determined by the customer's needs-based segment, the organization offers an enhanced one to the customer.

3. ______ the process of convincing a customer to stay with the organization at the point they are discontinuing service or convincing them to rejoin once they have left.

4. ______ the effort to win new, first-time customers.

Ⅳ. Answer the following questions.

1. Describe the advantages of customer relationship management (CRM).

2. Can you describe the four types of customer?

3. Explain the meaning of "cross sell" and "up sell" by example.

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

discontinue	trim	complementary
selectivity	screen	premium
innovative	filter	vulnerable

1. He was one of the most creative and ________ engineers of his generation.

2. I'll have to ________ these weekly visits.

3. The soldiers specialized in going out in small groups, to kill with a very high degree of ________.

4. The best prevention for cholera is to boil or ________ water, and eat only well-cooked food.

5. Grass shears are specially made to ________ grass growing in awkward places.

6. The airline had not been searching unaccompanied baggage by hand, but only ________ it on X-ray machines.

7. Plants that are growing vigorously are less likely to be ________ to disease.

8. Even if customers want "solutions", most are not willing to pay a ________ for them.

Collocation

VI. Look at the following sentence taken from the text. Just think about what else you can "**address**" and fill in the sentences with the right word.

Example: CRM allows a company to **address** all of the types of **customers** it serves at different points in their life cycle and to choose the marketing program that best fits a customer's attitude toward the company.

1. Ap________ should be addressed to: The business affairs editor.
2. He is due to address a co________ on human rights next week.
3. The two foreign ministers did not address each ot________ directly when they last met.
4. He addressed his re________ to Eleanor, ignoring Maria.
5. Mr. King sought to address those fe________ when he spoke at the meeting.
6. Throughout the book we have addressed ou________ to the problem of ethics.

Summary of the Text

VII. Choose the best one of the four choices given to fill in each blank.

CRM is the philosophy, policy and coordinating strategy connecting different players within an organization so __1__ to coordinate their efforts in creating an overall valuable series of experiences, products and services for the __2__. The objectives of a CRM strategy must consider a company's __3__ situation and its customers' needs and expectations. Information gained through CRM initiatives can support the development of marketing strategy by developing the organization's knowledge in areas __4__ identifying customer segments, improving customer retention, improving product offerings by better understanding customer needs, and by identifying the organization's most __5__ customers.

CRM strategies can __6__ in size, complexity and scope. Some companies consider a CRM strategy to only focus on the management of a __7__ of salespeople. However, other CRM strategies can cover customer __8__ across the entire organization. Many commercial CRM software packages that are __9__ provide features that serve sales, marketing, __10__ management,

project management and finance.

1. A. as	B. that	C. order	D. well
2. A. supplier	B. consumer	C. vender	D. customer
3. A. worse	B. specific	C. better	D. special
4. A. such as	B. by	C. like	D. as
5. A. particular	B. wealth	C. profitable	D. attractive
6. A. vary	B. various	C. range	D. divide
7. A. crowd	B. group	C. team	D. herd
8. A. expense	B. confidence	C. interaction	D. thought
9. A. available	B. bought	C. sold	D. systematic
10. A. event	B. incident	C. affair	D. order

Translation

Ⅷ. Translate the following into Chinese.

1. Because competitive advantage based on an innovative new product or service can be very short lived, companies are realizing that their only enduring competitive strength may be their relationships with there customers.

2. Besides affinity programs, organizations will often offer customized billing, special help lines or back-end loaded credits as a means of encouraging loyalty.

3. It increases each salesperson's efficiency in reducing the cost per sale as well as the cost of acquiring new customers and retaining old ones.

4. At levels lower than this the cost far outweighs the potential increase in gross profit.

5. In financial terms, when a customer accepts a cross-sell or up-sell offer, that customer begins to become much more profitable.

Chapter 19
E-Business and E-Commerce

Pre-reading Questions

1. What is E-Business?
2. Identify the benefits of E-Business.
3. What is E-Commerce?
4. What is the difference between E-Business and E-Commerce?

19.1 E-Business

1. Concept of E-Business

Electronic business (e-business) is the use of information technology and electronic communication networks to exchange business information and conduct transactions in electronic, paperless form.

E-business leverages the power of information technology and electronic communication networks, such as the Internet, to transform critical business strategies and processes. E-business removes traditional boundaries of time and geography and makes possible the creation of new virtual communities of suppliers and customers. As indicated in the definition, e-business includes the exchange of business information that may or may not directly relate to the purchase or sale of goods or services. For example, businesses are increasingly using electronic mechanisms to improve company performance by facilitating collaboration and data sharing among employees as well as to provide improved customer support. Participants in e-business transactions and information exchanges may be individuals (consumers and employees) or automated agents (information systems that are programmed to perform with little or no human intervention). Transactions and information exchanges can take place within a company, between companies, between companies and individuals, and between individuals. Another term commonly associated with e-business is e-commerce, which we define as the use of business to buy and sell products or services.

It is important to realize that e-business is not restricted just to "high-tech" companies. It is also important to understand that in the very near future there will not be a distinction between e-business and business in general. E-business is such a critical component of a successful business strategy that it will simply become a normal and universal aspect of business sometime in the future.

In fact, information technology and the Internet are facilitating business process re-engineering (BPR) on a large scale. In today's competitive and innovative environment, companies are being pushed to improve their existing business processes. Organizations looking to improve may examine their transactions to ensure they are serving key customers in an effective and efficient manner. Evidence of the importance of integrating e-business with traditional business processes is found in mergers and joint ventures between Internet companies such as America Online and Yahoo with traditional brick-and-mortar companies like Time Warner and Ford.

The DoorBuilder highlights the fact that companies wanting the most from e-business will take a good look at the way they are currently doing business. In many cases, for established companies to benefit most from e-business, they must re-engineer some of their processes. For example, in implementing e-business applications in sales and procurement, Sun Microsystems Inc. realized the power of both the Internet and of outsourcing operations to third parties. As a result — for the first time ever — Sun required every division chief to explain the division's core competencies and defend the need for personnel and resources to determine if it is in Sun's best interest to keep doing business as usual or move to an e-business solution developed in-house or outsourced.

2. Benefits of E-Business

The rapid growth in e-business clearly indicates that potential benefits greatly exceed costs. For example, e-business transactions are generally completed faster, cheaper, and more accurately than more traditional transactions, which may translate to a competitive edge. We now describe some of the potential benefits of e-business.

(1) Expanded Sales Opportunities

- Sales channels to new customers in all parts of the globe can be opened through e-business. For example, LandsEnd.com sells more clothing online than any other company. The industry-leading direct catalog merchant leveraged its customer service and fulfillment process to add a front-end Web site — a digital version of its well-known catalog. The Internet has allowed Lands' End to extend its channel to the rapidly growing global online community.
- A new storefront can be added to traditional brick-and-mortar companies. For example, Office Depot hosts the number one office-supply Web site, OfficeDepot.com. The Web site, which is profitable today, offers service and convenience without undercutting brick-and-mortar store prices.
- Providing product information, technical support, and order information online frees sales personnel from lower-value activities so they can pursue higher-value activities that generate new sales.
- All e-business can be opened all day, every day, and can be accessed from practically any

Internet connection.

- E-business can facilitate fast and flexible responses to market opportunities.

(2) Improved Communication, Customer Service, and Loyalty

As Internet traffic expands, a successful Web site will continue to be an important means of developing and maintaining corporate and brand images.

Company and product information is easily provided and maintained on the Internet.

Searchable electronic databases containing previously asked and answered questions can be useful for both consumers and customer support personnel.

Direct and customized product promotion is possible. For example, Ticketmaster.com E-mails customers the play list from the most recent concert they attended with an offer to sell a concert T-shirt.

The electronic information that often accompanies e-business can be a treasure chest of data useful in tracking customer patterns and other marketing information.

Online information and responses can be combined with human interaction. For example, ebank.com has a "call back" button on their Web site. When a user clicks the button and enters a phone number, software forwards the information to a call center, and an ebank.com representative calls the customer. Other sites offer live chat functions where customers can exchange instant messages with a customer service representative. According to Jupiter Communications, over 90 percent of online customers prefer some form of human interaction during the e-business experience.

To most customers, e-business means "business my way, on my time, in my place". Companies who are able to adapt quickly to satisfy this new type of customer will be successful. In a survey conducted by PricewaterhouseCoopers, creating customer loyalty is a top priority of business executives. These executives recognize there is substantial evidence that loyal customers are more likely to spend more, provide more feedback, and refer more business than less-devoted customers. Furthermore, loyal customers are expensive to replace — it is five times more costly to acquire new customers than to satisfy existing ones.

To stimulate and manage innovation, companies can set up "knowledge markets", which utilize various communication networks to connect small entrepreneurial teams that drive innovation. Once empowered, these small teams are relatively nimble and can function with a start-up mentality.

(3) Lower Costs

Procurement costs are lower.

Sales, marketing, and billing costs can be reduced. E-billing, file delivery of routine bills online, could save as much as 60 percent per bill. Most billers pay about $2 a month per customer in bill processing, printing, and postage. Savings in billing functions for a large utility could be as much as $50 million per year.

Hiring costs can be reduced. IBM has installed software that has cut the cost of hiring temporary workers by $3 million.

(4) Supply Chain and Human Resource Management

Proper supply chain management is critical to successful business. Companies try to avoid overstocking/inventory because it is expensive to store and overstocking increases the chance of obsolescence. On the other hand, companies want to have items in stock to fulfill customer orders quickly. E-business allows retailers, manufacturers, and suppliers to share forecasts and information about current inventory levels, which allows for significantly improved control over inventory levels.

Wider access to resources is available. General Electric Company bought $1 billion in supplies via the Internet in 1998. This saved the company 20 percent on materials costs because its divisions were able to reach a wider base of suppliers to negotiate better deals. In 2000, GE expects to spend $5 billion over the Internet.

Online buying groups can result in significant discounts. The Internet has spawned huge buying groups and consortiums that pool their corporate purchases to more better deals and special treatment. For example, Comdisco Inc., a computer refurbisher, buys an estimated $1 million worth of circuit breakers and wiring each year. The company hopes to save as much as $200000 by joining an online buying group purchasing equipment in huge volume. Vendors are happy to provide special treatment and quantity discounts because online groups lower vendor costs by supplying quick access to large, well-defined pools of buyers.

Response time with suppliers and vendors can be improved because many more of the interactions are automated. And even for those interactions that are not automated, transmission of an electronic document through the Internet (e.g., E-mail) is much faster than sending a physical document via the postal service or even an overnight courier ("snail mail").

Companies can leverage their purchasing power. IBM buys from suppliers who build only a small part of a machine. To ensure that IBM's various suppliers get IBM pricing from parts manufacturers, IBM wired 12000 suppliers to a network. IBM uses data-mining software to determine whether, for example, the price charged by a supplier for a particular computer chip is the same price IBM would pay for the chip if it were ordered directly from the manufacturer. If IBM can buy the chip at a lower price than can their supplier, IBM will go to the manufacturer and demand the same price for the supplier, who then passes along the savings to IBM.

Companies can improve their hiring and promotion processes. Home Depot recently automated its hiring and promotion processes by installing computer stations or kiosks in their stores. Job seekers apply for positions at the kiosks. The computer administers an extensive skills test and informs applicants when they are eligible for higher positions. For example, an applicant for a cashier position might demonstrate skills sufficient for a sales associate job; if so, the system makes this known. Applications are entered into a networked database so that managers at any Home Depot within commuting distance can access qualified personnel. Managers have reported that applicants tend to be more honest when completing applications at the kiosks compared with in-person interviews. Current employees also use the kiosks to apply for promotion. Managers consider at least 3 applicants for each position to be filled via promotion. The automated process has accomplished its

main purpose — to ensure that a broader pool of applicants are considered for jobs.

19.2 E-Commerce

1. Looking at E-Commerce from Different Perspectives

A three-layer model is commonly used to describe e-commerce such as, the one proposed by Zwass. Zwass's model consists of an infrastructure layer, a services layer, and a products/structures layer. The three layers can be further divided into seven functional layers for carrying out different functions. The major functions are to provide the:

- technical infrastructure (e.g. the Internet and www).
- secure messaging services (e.g. EDI).
- supporting services (e.g. electronic payment).

Greenstem and Feinman discuss another three-layer model consisting of the existing market space, the three pillars of e-commerce (electronic information, electronic relationships, and electronic transactions), and the open market processes. Kalakota and Robhlson view e-commerce from a wider perspective, using the term e-business. It is about integrating the frontend and backend applications with the business process. With the aim of maximizing customer value, it involves redefining the business model in conjunction with various information technologies. Eight business rules are proposed for achieving this goal. Schneider and Perry view e-commerce as an effective means to improve a value chain, which is used to link various functional activities (i.e. production, marketing, finance, etc.) of a company. This value chain concept can also be extended to link different companies to form an industry value chain. In general, e-commerce helps to facilitate information flow across the value chains and to reduce the associated transaction costs. It is also of interest to look at e-commerce from the point of view of relationship. At its root, every business needs to maintain three types of relationship: the relationship with its customer, the relationship with its business partners (e.g. suppliers), and the relationship with its employees. E-commerce provides an effective tool for building, managing, and enhancing these relationships. In the context of e-commerce, the first type of relationship is not just selling through the web but managing customer relationships in general. Special electronic customer relationship management software is available for this purpose. The second type of relationship is about procurement and supply chain management by electronic means. Virtual Private Networks and XML are the main facilitators in these areas. While the first two types of relationship are external, the last one is internal. It involves building an effective Intranet for integrating different information systems and sharing information through which communication and productivity can be enhanced.

2. Different Types of E-Commerce

The matrix in Table 19-1 shows the different types of e-commerce from the perspective of the buyer and seller relationship. This is often used to categorize e-commerce applications. According

to this relationship, e-commerce applications can be divided into the following four categories.

Table 19-1　Different Types of E-Commerce

	Business (organization)	Consumer (individual)
Business (organization)	B2B	B2C
Consumer (individual)	C2B	C2C

- Business-to-Consumer (B2C). In this case, the seller is a business organization whereas the buyer is a consumer. This emulates the situation of physical retailing and so it is commonly called electronic retailing. Typically, electronic stores are set up on the Internet to sell goods to the consumers. For example, our VBS sells books to the consumers through the Internet. Note here that the business drives the specification of the product and the customer chooses whether or not to buy a prefabricated product. An example of this in traditional commerce is purchasing suits "off the rack".
- Business-to-Business (B2B). In this case, both the buyer and the seller are business organizations. There are three types of systems, namely, buyer-oriented system, seller-oriented system, and virtual marketplace. In many situations, it is related to supply chain management. For example, the Virtual Bookstore (VBS) needs to order books from various publishers. The ordering process can be accomplished by using electronic data interchange.
- Consumer-to-Consumer (C2C). This refers to situations where both the seller and the buyer are consumers. With the advent of e-commerce, on-line auctions provide an effective means for supporting (C2C) e-commerce. For example, our VBS can provide on-line auction services for customers to sell used books to other customers through the VBS web site. In addition, a virtual community can be formed.
- Consumer-to-Business (C2B). As explained later, this is a new form e-commerce in which a consumer specifies the requirements to a business, which provides a product that meets these requirements. These requirements could be as simple as an acceptable price, or could involve considerable customization of an existing standard product, or creation of a new product. An example of this in the traditional commerce setting is a "made to measure" tailor. The key distinction is related to who is driving the specification of the product being purchased. Unlike B2C, there is a strong element of customization.

Technical Notes to the Text

1. electronic business (EB)，电子商务。电子商务是运用以 Internet 等为代表的现代信息技术进行所有与商务有关的活动，如商务信息、商务管理、商品交易等。

2. electronic commerce (EC)，电子业务。电子业务指的是利用简单、快捷、低成本的电子通信方式，买卖双方不谋面地进行各种商贸活动。

3. business process reengineering (BPR)，业务流程重组。企业流程重组工程是对企业的

业务流程做根本性的思考和彻底重建，其目的是在成本、质量、服务和速度等方面取得显著的改善，使企业能最大限度地适应以顾客(customer)、竞争(competition)、变化(change)为特征的现代企业经营环境。

4. electronic data interchange (EDI)，电子数据交换。电子数据交换指利用符合标准的结构化的信息从计算机到计算机之间的电子传输。电子数据交换是一种商业贸易的工具，将商业文件(如订单、发票、货运单、报关单和进出口许可证)按统一的标准编制成计算机能识别和处理的数据格式，在计算机之间进行传输。电子数据交换俗称“无纸贸易”。

5. B2B (business to business)，企业到企业电子商务，也称为商家对商家或商业机构对商业机构电子商务，即 B to B。商业机构对商业机构的电子商务是指商业机构(或企业、公司)使用 Internet 或各种商务网络向供应商(企业或公司)订货和付款。这种模式的电子商务发展最快，已经有了多年的历史，特别是通过增值网络(value added network，VAN)上运行的电子数据交换，使企业对企业的电子商务得到了迅速扩大和推广。公司之间可能使用网络进行订货、签合同等单证和付款。

6. B2C (business to consumer)，企业到消费者电子商务，也称商家对个人客户或商业机构对消费者电子商务，即 B to C。商业机构对消费者的电子商务基本等同于电子零售商业。目前，Internet 上已遍布各种类型的商业中心，提供各种商品和服务，主要有鲜花、书籍、服装、计算机、汽车等商品和服务。

7. supply chain，供应链。供应链是产品生产和流通过程中所涉及的原材料供应商、生产商、批发商、零售商，以及最终消费者组成的供需网络。供应链管理(supply chain management，SCM)是一种跨行业的管理，是指人们在认识和掌握了供应链各环节内在规律和相互联系的基础上，利用管理的计划、组织、指挥、协调、控制和激励职能，对产品生产和流通过程中各个环节所涉及的物流、信息流、资金流、价值流及业务流进行的合理调控，以期达到最佳组合，发挥最大效率，迅速以最小的成本为客户提供最大的附加值。

8. transaction processing systems (TPS)，事务处理系统。TPS 是使操作层的信息处理自动化，从而提高工作效率的信息系统。事务处理是组织最基本的活动，把信息按照事务组织和处理。实现组织日常事务活动的系统称为事务处理系统。其主要作用是反馈控制。

9. electronic funds transfer (EFT)，电子资金转账。电子转账系统是银行同其客户进行数据通信的工具，是一个采用计算机和通信线路的电子银行系统，用户传输同金融交易有关的电子资金和相关的数据与信息，一般具有支付功能和信息服务功能。

10. neural network，神经网络。它是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。

Word Bank to the Text

A. Useful new words

leverage *n.* 杠杆作用

collaboration	*n*. 协作
merger	*n*. 合并，归并
established	*adj*. 确立的，确认的
procurement	*n*. 收买，采购
outsourcing	*n*. 外部采办，外购
competency	*n*. 资格，能力
storefront	*n*. 店面，铺面
undercut	*v*. 削弱
promotion	*n*. 促进，提升，提拔
substantial	*adj*. 坚固的，实质的
innovation	*n*. 改革，创新
entrepreneur	*n*. 企业家，创业者
empower	*v*. 授权
nimble	*adj*. 敏捷的
mentality	*n*. 智力，精神
overstock	*v*. 进货过多　*n*. 库存过剩
spawn	*v*. 造成，使产生
consortium	*n*. 社团，协会
courier	*n*. 信使，送急件的人
kiosk	*n*. 书报亭，报摊
eligible	*adj*. 适任的，合格的
commute	*v*. 定期往返于两地间
emulate	*v*. 仿效
prefabricate	*v*. 预制
auction	*n*. 拍卖

B. Useful expressions

with little or no intervention	没有介入
be restricted to	限制
on a large scale	大规模的
benefit from	受益
extend its channel	拓宽渠道
exchange messages with	交换信息
conduct a survey	进行调查
increase the chance of	提高机会
have sth. in stock	库存
worth of	价值
in huge volume	大量的

be eligible for	有资格
in conjunction with	和……一道，结合
from the point view of	从……观点来看
off the rack	现成的

C. Technical terms and proper names

electronic business (EB)	电子商务
electronic commerce (EC)	电子业务
business process reengineering (BPR)	业务流程重组
electronic data interchange (EDI)	电子数据交换
intranet value chain	企业内联网价值链
B2B (business to business)	企业到企业
B2C (business to consumer)	企业到消费者
C2C (consumer to consumer)	消费者到消费者
C2B (consumer to business)	消费者到企业
supply chain	供应链
human resource management	人力资源管理

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. Zwass's three-layer model consists of an ________ layer, a services layer, and a _________ layer.

2. E-business removes traditional ________ of time and geography and makes possible the creation of new virtual ________ of suppliers and customers.

3. The rapid growth in e-business clearly indicates that _______ benefits greatly exceed ________.

4. Searchable electronic ________ containing previously asked and answered questions can be useful for both consumers and customer support personnel.

5. Business-to-consumer (B2C). In this case, the seller is a ________ whereas the buyer is a ________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. Greenstem and Feinman's three-layer model consists of an infrastructure layer, a services layer, and a products/structures layer. (   )

2. Online information and responses can be combined with human interaction. (　　)

3. In the case of Business-to-Consumer (B2C), the seller is a consumer whereas the buyer is business organization. (　　)

4. Internet based e-business is as expensive as paper-based transactions or even traditional electronic data interchange (EDI). (　　)

5. Response time with suppliers and vendors can be improved because many more of the interactions are automated. (　　)

Ⅲ. Match each of the following terms with the appropriate definition.

Electronic business	Business-to-Business (B2B)
Consumer-to-Consumer (C2C)	Consumer-to-Business (C2B)

1. ______ the use of information technology and electronic communication networks to exchange business information and conduct transactions in electronic, paperless form.

2. ______ This refers to situations where both the seller and the buyer are consumers.

3. ______ In this case, both the buyer and the seller are business organizations.

4. ______ a consumer specifies the requirements to a business, which provides a product that meets these requirements.

Ⅳ. Answer the following questions.

1. Distinguish between E-business and E-commerce.
2. Explain how E-business can improve communication, customer service, and loyalty.
3. Describe the risk that faces in E-business.
4. Distinguish B2B, B2C, C2C, C2B.

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

Auction	undercut	empower
Customization	promotion	eligible
commute	collaboration	innovation

1. He was also a writer of beautiful stories, some of which are ________ with his fiancee.
2. The firm will be able to ________ its competitors while still making a profit.
3. Advertising is often the most effective method of ________.
4. We must promote originality, inspire creativity and encourage ________.
5. What I'm trying to do is to ________ people, to give them ways to help them get well.
6. Almost half the population are ________ to vote in today's election.
7. She ________ from Oxford to London every day.

8. The government is taking emergency ________ to deal with a housing crisis.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can “conduct” and fill in the sentences with the right word.

Example: Electronic business (e-business) is the use of information technology and electronic communication networks to exchange business information and **conduct transactions** in electronic, paperless form.

1. The duty of every management is to conduct the bu______, including the price policy of the business.
2. Plastic and rubber won’t conduct el______, but copper will.
3. With voice mail, you can conduct entire co______ without ever reaching anyone.
4. Even today, some people conduct ex______ on lab animals to find possible approaches to remain young.
5. Today, corporations hardly ever introduce major new products without conducting elaborate market re______.
6. He said they were conducting a ca______ against democrats across the country.

Summary of the Text

Ⅶ. Choose the best one of the four Choices given to fill in each blank.

Electronic Business may be defined broadly as any business process that relies on an automated information system. Today, this is __1__ done with Web-based technologies. E-business methods __2__ companies to link their internal and __3__ data processing systems more efficiently and flexibly, to work more closely with suppliers and __4__, and to better satisfy the needs and expectations of their customers.

In practice, e-business is __5__ than just e-commerce. While e-business refers to more strategic focus with an emphasis __6__ the functions that occur using electronic capabilities, e-commerce is a subset of an __7__ e-business strategy. E-commerce seeks to add revenue streams using the World Wide Web or the Internet to build and enhance relationships with clients and partners and to improve efficiency using the Empty Vessel strategy. Often, e-commerce __8__ the application of knowledge management systems.

It is important to realize that e-business is not __9__ just to high-tech companies. It is also important to understand that in the very near future there will not be a __10__ between e-business and business in general.

1. A. most	B. much	C. more	D. mostly
2. A. enables	B. unable	C. able	D. enable
3. A. internal	B. interior	C. external	D. exterior

4. A. customer	B. consumers	C. partners	D. providers
5. A. more	B. much	C. a lot	D. great
6. A. on	B. in	C. about	D. over
7. A. over	B. overall	C. part of	D. all
8. A. involves	B. refers	C. means	D. prefers
9. A. implied	B. covered	C. restricted	D. closed
10. A. distinction	B. link	C. association	D. connection

Translation

Ⅷ. Translate the following into Chinese.

1. As indicated in the definition, e-business includes the exchange of business information that may or may not directly relate to the purchase or sale of goods or services.

2. Evidence of the importance of integrating e-business with traditional business processes is found in mergers and joint ventures between Internet companies.

3. As Internet traffic expands, a successful Web site will continue to be an important means of developing and maintaining corporate and brand images.

4. To most customers, e-business means "business my way, on my time, in my place".

5. Companies try to avoid overstocking/inventory because it is expensive to store and overstocking increases the chance of obsolescence.

Chapter 20
Artificial Intelligence

Pre-reading Questions

1. What is artificial intelligence?
2. What is the relationship between artificial intelligence and machine leaning?
3. How many machine leaning methods do you know? Name at least four of them.

The ultimate goal of artificial intelligence is to mimic human-like "cognitive" functions in machine, which can be accomplished through learning algorithms that mimic the human-like leaning. Thus, machine learning is one of the most exciting recent technologies in artificial intelligence as it enables the machines to gain human-like intelligence. In this chapter, we will introduce history of AI, basics of AI and applications of AI in machine learning.

20.1 History of AI

Artificial intelligence (AI), sometime called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals. Specifically, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

The field of AI was formally founded as an academic discipline in 1956, at a conference at Dartmouth College, in Hanover, New Hampshire, where the term "artificial intelligence" was coined. Attendees Allen Newell (Carnegie Mellon University), Herbert Simon (Carnegie Mellon University), John McCarthy (Massachusetts Institute of Technology), Marvin Minsky (Massachusetts Institute of Technology) and Arthur Samuel (International Business Machines Corporation (IBM)) became the founders and leaders of AI research.

After experiencing "AI winter" — a period of disappointment and the loss of funding from 1974 to 1980, the field later revived in the 1980s when the British government started funding it again in part to compete with efforts by the Japanese. The field experienced another major winter from 1987 to 1993, coinciding with the collapse of the market for some of the early general-purpose computers, and reduced government funding.

But research began to pick up again after that. In the early 1980s, AI research was revived by the commercial success of expert systems, a form of AI program that simulated the knowledge and analytical skills of human experts. In the late 1990s and early 21st century, AI

began to be used for logistics, data mining, medical diagnosis and other areas. The success was due to increasing computational power, greater emphasis on solving specific problems, new ties between AI and other fields (such as statistics, economics and mathematics), and a commitment by researchers to mathematical methods and scientific standards. Faster computers, algorithmic improvements, and access to large amounts of data enabled advances in machine learning and perception; data-hungry deep learning methods started to dominate accuracy benchmarks around 2012. Until today, AI techniques have experienced a resurgence and become an essential part of the technology industry, helping to solve many challenging problems in computer science.

Many achievements had been made during this period. In 1997, IBM's Deep Blue became the first computer chess-playing system to beat a reigning world chess champion, Garry Kasparov. In 2011, the computer giant's question-answering system Watson won the quiz show "Jeopardy!" by beating reigning champions Brad Rutter and Ken Jennings. In 2016, AlphaGo won 4 out of 5 games of Go in a match with Go champion Lee Sedol, becoming the first computer Go-playing system to beat a professional Go player without handicaps. In the 2017 Future of Go Summit, AlphaGo won a three-game match with Ke Jie, who at the time continuously held the world No.1 ranking for two years. This marked the completion of a significant milestone in the development of Artificial Intelligence as Go is an extremely complex game, more so than Chess.

20.2　Basics of AI

AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. The traditional problems of AI research include reasoning, knowledge representation, planning, problem solving, learning, natural language processing, perception and the ability to move and manipulate objects. Many tools are used in AI, including versions of search and mathematical optimization, artificial neural networks, and methods based on statistics, probability and economics. The AI field draws upon computer science, mathematics, psychology, linguistics, philosophy and many others.

Knowledge engineering is a core part of AI research. Machines can often act and react like humans only if they have abundant information relating to the environment. AI must have access to objects, categories, properties and relations between all of them to implement knowledge engineering. Initiating common sense, reasoning and problem-solving power in machines is a difficult and tedious task.

Machine learning is another core part of AI. It investigates how computer agents can improve their perception, cognition, and action with data, knowledge, experience, and interaction. Machine learning utilizes a variety of techniques to intelligently handle large and complex amounts of information build upon foundations in many disciplines, including statistics,

knowledge representation, databases, causal inference, computer systems, machine vision, natural language processing and so on.

Machine perception deals with the capability to use sensory inputs to deduce the different aspects of the world. For example, computer vision is the power to analyze visual inputs with a few sub-problems such as facial, object and gesture recognition.

Robotics is also a major field related to AI. Robotics is a field of engineering focused on the design and manufacturing of robots. Robots require intelligence to handle tasks such as object manipulation and navigation, along with sub-problems of localization, motion planning and mapping.

20.3 Applications of Artificial Intelligence in Machine Learning

1. Introduction

Machine learning had grown out of the field of AI. More specifically, AI perceives environment and takes actions through AI program and machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data. It is of utmost importance as it enables the machines to gain human-like intelligence without explicit programming. Nowadays, machine learning has become one of the most exciting recent technologies in AI.

2. Machine Learning

The term "machine learning" was coined in 1959 by Arthur Samuel. According to Arthur Samuel, machine learning is defined as the field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel was famous for his checkers playing program. Initially when he developed the checkers playing program, Arthur was better than the program. But over time the checkers playing program beat Arthur through learning.

A more formal definition was given by Tom Mitchell as a computer program is said to learn from experience (E) with respect to some task (T) and some performance measure (P), if its performance on T, as measured by P, improves with E then the program is called a machine learning program.

In general, mathematical analysis of machine learning algorithms and their performance is a well-defined branch of theoretical computer science which often referred to as computational learning theory.

Until today, a diverse array of machine learning algorithms has been developed to cover the wide variety of applications and problems. Machine learning algorithms are typically classified into four broad categories.

- Supervised Learning

This learning process is based on the comparison of computed output and expected output. The expected outputs are labeled, and the correct answers are known. The algorithm refers to computing the error and adjusting the error for achieving the expected output. Supervised learning problems can be further grouped into regression and classification problems. For example, a data set of houses of particular size with actual prices is given, then the regression algorithm is to produce more of these right answers for new houses.

- Unsupervised Learning

Unsupervised learning is termed as learned by its own that discoveries and adopts the pattern based on the input. In this learning, the data are divided into different clusters and hence the learning is called a clustering algorithm. For example, clustering can be used to group all the shopping items available on the web into a set of unique products.

- Reinforcement Learning

Reinforcement learning is based on output with how an agent ought to take actions in an environment so as to maximize some notion of long-term reward. A reward is given for correct output and a penalty for wrong output. Reinforcement learning differs from the supervised learning problem in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected.

- Recommender Systems

Recommender systems can be defined as a learning technique by virtue of which online user can customize their sites to meet customer's tastes. For example, online users can get a rating of a product or/ and related items when he/she is searching items because of the existing recommender system. It changed the way that people find products, information, and even other people. There are mainly two approaches: content based recommendation and collaborative recommendation, which help the user for obtaining and mining data, making intelligent and novel recommendations. Most e-commerce site uses this system.

3. Approaches in Machine Learning

Machine learning has strong ties to mathematical optimization, which delivers methods and theories domain to the field. Machine learning methods are used to devise complex analytical models and algorithms, and these analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Machine learning algorithms vary greatly to cover the wide variety of data types, problem types performance metrics, computational complexity and so on. For example, some algorithms represent candidate programs (e.g., decision trees, mathematical functions, and general programming languages). Some algorithms focus on function approximation problems (e.g., given an input transaction, out-put a "fraud" or "not fraud" label), and the learning problem is to improve the accuracy of that function. In the area of machine learning research, the emphasis is

given more on choosing or developing an algorithm and conducting experiments on the basis of real world applications. Here, we present several typical examples of machine learning approaches.

Examples of Approaches

(1) Decision Tree Learning

Decision tree learning uses a decision tree as a predictive model, which maps observations about an item to conclusions about the item's target value.

(2) Artificial Neural Networks

An artificial neural network learning algorithm, usually called "neural network", is a learning algorithm that is vaguely inspired by biological neural networks. Computations are structured in terms of an interconnected group of artificial neurons, processing information using a connectionist approach to computation. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs, to find patterns in data, or to capture the statistical structure in an unknown joint probability distribution between observed variables.

(3) Deep Learning

Falling hardware prices and the development of Graphics Processing Units (GPUs) for personal use in the last few years have contributed to the development of the concept of deep learning which consists of multiple hidden layers in an artificial neural network. This approach tries to model the way the human brain processes light and sound into vision and hearing. Some successful applications of deep learning are computer vision and speech recognition.

(4) Support Vector Machines

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.

(5) Clustering

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to some predesignated criterion or criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness (similarity between members of the same cluster) and separation between different clusters. Other methods are based on estimated density and graph connectivity. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis.

(6) Bayesian Networks

A Bayesian network, belief network or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independencies

via a directed acyclic graph. For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning.

(7) Genetic Algorithms

A genetic algorithm (GA) is a search heuristic that mimics the process of natural selection, and uses methods such as mutation and crossover to generate new genotype in the hope of finding good solutions to a given problem. In machine learning, genetic algorithms found some uses in the 1980s and 1990s. Conversely, machine learning techniques have been used to improve the performance of genetic and evolutionary algorithms.

(8) Rule-based Machine Learning

Rule-based machine learning is a general term for any machine learning method that identifies, learns, or evolves “rules” to store, manipulate or apply, knowledge. The defining characteristic of a rule-based machine learner is the identification and utilization of a set of relational rules that collectively represent the knowledge captured by the system. This is in contrast to other machine learners that commonly identify a singular model that can be universally applied to any instance in order to make a prediction. Rule-based machine learning approaches include learning classifier systems, association rule learning, and artificial immune systems.

(9) Learning Classifier Systems

Learning classifier systems are a family of rule-based machine learning algorithms that combine a discovery component (e.g. typically a genetic algorithm) with a learning component (performing either supervised learning, reinforcement learning, or unsupervised learning). They seek to identify a set of context-dependent rules that collectively store and apply knowledge in a piecewise manner in order to make predictions.

(10) Feature Selection Approach

Feature selection is the process of selecting an optimal subset of relevant features for use in model construction. It is assumed that the data contains some features which are either redundant or irrelevant, and can thus be removed to reduce calculation cost without incurring much loss of information. Common optimality criteria include accuracy, similarity and information measures.

Technical Notes to the Text

1. artificial intelligence，人工智能。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

2. machine learning，机器学习。它是一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。专门研究计算机怎样模拟或实现人类的学习

行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能。

3. artificial neural network，人工神经网络。它从信息处理角度对人脑神经元网络进行抽象，建立某种简单模型，按不同的连接方式组成不同的网络。在工程与学术界也常直接简称为神经网络或类神经网络。

4. deep learning，深度学习。深度学习源于人工神经网络的研究，含多隐藏层的多层感知器就是一种深度学习结构。深度学习通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示。

5. expert system，专家系统。专家系统是一个智能计算机程序系统，其内部含有大量的某个领域专家水平的知识与经验，能够利用人类专家的知识和解决问题的方法来处理该领域问题。

6. big data，大数据。大数据指的是所涉及的资料量规模巨大到无法通过目前主流软件工具在合理时间内达到撷取、管理、处理，并整理成为帮助企业经营决策更积极目的的资讯。

7. data mining，数据挖掘。它是数据库知识发现中的一个步骤。数据挖掘一般是指从大量的数据中通过算法搜索隐藏于其中信息的过程。数据挖掘通常与计算机科学有关，并通过统计、在线分析处理、情报检索、机器学习、专家系统(依靠过去的经验法则)和模式识别等诸多方法来实现上述目标。

8. knowledge engineering，知识工程。这门新兴的工程技术学科产生于社会科学与自然科学的相互交叉和科学技术与工程技术的相互渗透。

9. supervised learning，监督学习。其是利用一组已知类别的样本调整分类器的参数，使其达到所要求性能的过程，也称为监督训练或有教师学习。

10. unsupervised learning，无监督学习。其用于在大量无标签数据中发现信息。它的训练数据是无标签的，训练目标是能对观察值进行分类或者区分等。

11. reinforcement learning，强化学习。它是近年来机器学习和智能控制领域的主要方法之一。强化学习关注的是智能体如何在环境中采取一系列行为，从而获得最大的累积期望回报。

12. recommender system，推荐系统。它是根据用户的特征，如兴趣爱好，推荐满足用户要求的对象，也称个性化推荐系统。

13. decision tree，决策树。其指在已知各种情况发生概率的基础上，通过构成决策树来求取净现值的期望值大于等于零的概率，评价项目风险，判断其可行性的决策分析方法，是直观运用概率分析的一种图解法。由于这种决策分支画成图形很像一棵树的枝干，故称决策树。

14. biological neural network，生物神经网络。生物神经网络一般指由生物的大脑神经元、细胞、触点等组成的网络，用于产生生物的意识，帮助生物进行思考和行动。

15. support vector machine，支持向量机。它是一种常见的判别方法。在机器学习领域，是一个有监督的学习模型，通常用来进行模式识别、分类及回归分析。

16. classification，分类。首先从数据中选出已经分好类的训练集，在该训练集上运用数据挖掘分类的技术，建立分类模型，对没有分类的数据进行分类。

17. regression，回归。回归是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。

18. clustering，聚类。将物理或抽象对象的集合分成由类似的对象组成的多个类的过程

称为聚类。由聚类所生成的簇是一组数据对象的集合，这些对象与同一个簇中的对象彼此相似，与其他簇中的对象相异。

19. bayesian networks，贝叶斯网络。其是描述数据变量之间依赖关系的一种图形模式，是一种用来进行推理的模型。贝叶斯网络为人们提供了一种方便的框架结构来表示因果关系，这使得不确定性推理变得在逻辑上更为清晰，可理解性强。

20. genetic algorithms，遗传算法。它是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

21. learning classifier systems，学习分类器系统。它是数据挖掘的一种非常重要的方法。分类的概念是在已有的数据基础上学会一个分类函数或构造出一个分类模型，即我们通常所说的分类器(classifier)。该函数或模型能够把数据库中的数据记录映射到给定类别中的某一个，从而可以应用于数据预测。

Word Bank to the Text

A. Useful new words

implement	*v.* 实施，执行，使生效
stream	*n.* 流，一连串，水流方向，小河
ultimate	*adj.* 极限的，最后的，最大的，首要的
mimic	*v.* 模拟，模仿
resurgence	*n.* 回潮，再起，复苏，中断之后的继续
cognitive	*adj.* 认知的，认识的
revive	*v.* 使复活，使恢复，使再生，使重新流行
vaguely	*adv*.含糊地，茫然地，暧昧地
benchmark	*n.* 基准，参照，标准检查程序
handicap	*v.* 妨碍，使不利
perceive	*v.* 理解，意识到，察觉，发觉
deduce	*v.* 演绎，推论，推断，追溯根源
tractable	*adj.* 易处理的，驯服的，温顺的
prominence	*n.* 突出，声望，卓越
connectionist	*n.* 连接
multiple	*adj.* 多重的，多个的，复杂的，多功能的
acyclic	*adj.* 非循环的，非周期的
heuristic	*adj.* 启发式的，探试的，探索的
piecewise	*adv.* 分段地

B. Useful expressions

in contrast to	相比之下
pick up	捡起，接载，学会，逮捕

match with	与……相配，使与……较量，与……一致
a large amount of	大量，一大堆
draw upon	利用，使用，采用
grow out of	产生自……，从……发展，渐渐穿不上，停止
define as	界定，定义为
with respect to	关于，(至于)谈到
be based on	以……为基础
refer to	参考，指的是，涉及，适用于
in general	通常，大体上，一般而言，总的来说
be famous for	因……而著名
divide into	分成，分为
ought to	理应，应当，应该，宜
take action	采取行动，行动起来
by virtue of	凭借……的力量，由于
give for	牺牲，交换
consist of	包括，由……组成
mark as	把……定为
belong to	属于，是(某团体、国家等)的成员，归于
seek to	设法，企图，试图

C. Technical terms and proper names

artificial intelligence (AI)	人工智能
machine learning	机器学习
artificial neural network (ANN)	人工神经网络
deep learning	深度学习
expert system	专家系统
big data	大数据
data mining	数据挖掘
knowledge engineering	知识工程
the game of go	围棋
supervised learning	监督学习
unsupervised learning	无监督学习
reinforcement learning	强化学习
recommender system	推荐系统
decision tree	决策树
support vector machine (SVM)	支持向量机
classification	分类
regression	回归
clustering	聚类

e-commerce	电子商务
bayesian network	贝叶斯网络
genetic algorithm	遗传算法
learning classifier system	学习分类器系统
computer science	计算机科学
joint probability distribution	联合概率分布
graphics processing unit (GPU)	图形处理器
intelligent agent	智能体
hidden layer	隐藏层
graph connectivity	图连通
technology industry	科技产业
natural language processing	自然语言处理
medical diagnosis	医疗诊断
data-hungry	大流量数据
cluster	簇
feature selection approach	特征选择方法
rule-based machine learning	基于规则的机器学习
variable	变量

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. __________ mimics human-like brain cognitive functions and ____________ mimics human-like leaning.

2. Supervised learning problems includes _________ and _________ problems.

3. __________ is the method that assigns of a set of observations into subsets, where observations within the same set are similar, while observations drawn from different set are dissimilar.

4. Generally, machine learning algorithms can be divided into four categories:__________, __________, __________ and __________.

5. List four machine leaning approaches:__________, __________, __________, __________ And __________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. The term "artificial intelligence" was coined by Arthur Samuel.　　(　　)

2. Machine perceives the environments by using sensors. ( )

3. In supervised learning, we use unlabeled data. ( )

4. "Big Data" has become dominant in machine learning algorithms. ( )

5. An artificial neural network learning algorithm is inspired by biological neural networks. ( )

Ⅲ. Match each of the following terms with the appropriate definition.

genetic algorithm (GA)	clustering
AlphaGo	robotics

1. _______ A field related to AI that requires intelligence agent to perform tasks such as object manipulation and navigation.

2. _______ A computer program that plays the board game Go.

3. _______ A method that group set of objects in such a way that objects in the same group are more similar to each other than to those in other groups.

4. _______ A method mimics the process of natural selection, and it generates new genotype to find good solutions to a given problem.

Ⅳ. Translate the following into Chinese.

1. speech recognition 2. biological neural network 3. computer vision
4. mathematical optimization 5. random variable

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

implement	mimic	ultimate
benchmark	handicap	prominence
perceive	deduce	multiple

1. Robots ______ the environment with sensors that have both geometric and physical limitations.

2. The ______ for artificial intelligence is that it lacks the pattern recognition tools.

3. Genetic algorithms are programming techniques that ______ biological evolution as a problem-solving strategy.

4. ______ regression is regression analysis with more than one independent variable.

5. Deep learning, a further subset of machine learning gaining lot of ______ recently, imitates the workings of the human brain in processing data and creating patterns for use in decision making.

6. A standard ________ used by computer vision researchers has shown a fourfold improvement in image classification accuracy from 2010 to 2014.

7. In decision tree, these ______ branches of the tree are called leaf nodes.

8. Today, the power necessary to ________ these advanced system designs is readily available.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can "improve" and fill in the sentences with the right word.

Example: Machine learning methods applied to DNA microarray data can **improve** the **diagnosis** of cancer.

1. How to improve the quality of pro______ was a difficult problem to him.

2. The report goes on to make a number of recommendations to improve saf______ on aircraft.

3. For big companies, outsourcing part of their manufacturing business can help cut costs and improve effi______.

4. Today, more than ever, we need the public understanding and engagement of science. It is essential for science to influence policy and improve people's li______.

5. A study has found that the simple act of walking may improve mem______ in old age.

6. We will introduce a simple example of reading a shopping cart and analyzing it to find patterns that improve perf______.

Summary of the Text

Ⅶ. Choose the best one of the four Choices given to fill in each blank.

This chapter introduces history of AI, basics of AI and applications of AI in machine learning. AI technologies enable machines to mimic "cognitive" functions as human minds, __1__ as "learning" and "problem __2__ ". Knowledge engineering, machine learning, machine perception and robotics are the core parts of AI. AI research is highly related to other fields, such as statistics, economics, mathematics and so on. In the twenty-first century, AI techniques have developed with __3__ speed following advances in computer power, large __4__ of data, and theoretical understanding. __5__, AI techniques have become an essential part of the technology industry, helping to solve many challenging problems in computer science. Machine learning is one of the most __6__ recent technologies in AI, __7__ it enables machine to mimic human-like leaning and finally __8__ the human-like intelligence. Various machine learning algorithms have been designed to __9__ the wide variety of data types, problem types performance metrics, computational complexity and so on. __10__, machine learning algorithms are classified into four broad categories: supervised learning, unsupervised learning, reinforcement learning, recommender systems. The algorithm choosing and developing are based on the real world applications.

1. A. sound　　B. well　　C. act　　D. such

2. A. solve	B. solving	C. solution	D. solved
3. A. low	B. quickly	C. high	D. decrease
4. A. amounts	B. number	C. sum	D. size
5. A. Also	B. Thus	C. Although	D. However
6. A. inhumane	B. exciting	C. disappoint	D. boring
7. A. although	B. and	C. because	D. however
8. A. return	B. gain	C. proceed	D. discover
9. A. identify	B. explain	C. cover	D. include
10. A. Commonly	B. Certainly	C. Actually	D. Probably

Translation

Ⅷ. Translate the following into Chinese.

1. Artificial intelligence is a branch of computer science that aims to create intelligent machines. It has become an essential part of the technology industry.

2. Breakthroughs should be made in basic theories of AI, such as big data intelligence, multimedia aware computing, human-machine hybrid intelligence, swarm intelligence and automated decision-making.

3. AI has gained prominence recently due, in part, to big data, or to the increase in speed, size and the variety of data business are now collecting.

4. AI can perform tasks such as identifying patterns in the data more efficiently than humans, enabling businesses to gain more insight out of their data.

5. Machine learning is the science of getting a computer to act without programming. Deep learning is a subset of machine learning that, in very simple terms, can be thought of as the automation of predictive analytics.

Chapter 21

Big Data and Cloud Computing

Pre-reading Questions

1. What is big data?

2. What is cloud computing?

3. Are you familiar with the industries that benefit from big data application? Name at least four of them.

Big data and cloud computing are the two most trending terms which have gained much attention from academia and information technology industry today. Big data represents so huge quantity of data that is virtually impossible to process by just one machine and can be further processed to extract information. Cloud computing is an application that systematically stores data and programs using a network of remote servers over the Internet. The combination of the two provides a solution to the challenge of large volumes of data today. In this chapter, we will introduce background, big data and cloud computing.

21.1 Background

Big data is an umbrella term which encompasses all sorts of data existing today. From hospital records to the overwhelming amounts of information in social network — there is more to it than we officially know. These data can be further processed to extract information. It provides keen insights to the prospective owners who would then gather, store, and organize the data for further analysis.

Cloud computing provides the infrastructure for gathering data and information over the Internet. Rather than a personal computer or a local server, it utilizes a host of remote servers to manage and process huge volumes of data. It provides the platform to share computer facilities to run programs.

Nowadays, the cloud is growing steadily, and the information across web is also exploding. For example, in social media, both marketers and common users generate loads of data everyday. Organizations and institutions are also creating data on a daily basis. Currently, over 2 billion people worldwide are connected to the internet, and there are over 5 billion individuals owning mobile phones. By 2020, 50 billion devices are expected to be connected to the Internet. The high volumes of data present a challenge to the cloud environment, which can eventually become difficult to manage. It is necessary to find a way to manage the data rather than just stacking it.

Big data and cloud computing techniques are changing the way of collecting, analyzing, storing, and using data. The combination of the two provides a solution which is both scalable and accommodating for huge data.

Big data and cloud computing are the two most trending terms in the ever-growing information technology (IT) world today. This trend not only becomes the reality with a far-reaching influence on business, but also big data has reached its limits to deploy services to the IT infrastructure.

21.2 Big Data

1. Definition of Big Data

As a result of this technological revolution, these millions of people are generating tremendous amounts of data through the increasing use of such devices. In particular, remote sensors continuously produce much heterogeneous data that is either structured or unstructured. This data is known as big data. Big data is so huge and complex that traditional data-processing application software are inadequate to deal with them. In the IT industry as a whole, the rapid rise of Big Data has generated new issues and challenges with respect to data capturing, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy and data source.

Nowadays, big data is still in its infancy stage, and the domain has not been reviewed in general. The term "big data" has been in use since the 1990s, with some giving credit to John Mashey for coining or at least making it popular.

Lately, the term "big data" tends to refer to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. For example, to manage a factory, one must consider both visible and invisible issues with various components. Information generation algorithms must detect and address invisible issues such as machine degradation, component wear, and so on.

2. Characteristics

There are a number of concepts associated with big data. Originally, big data can be described by 3 characteristics: volume, variety, velocity. Other characteristics later attributed with big data are veracity and value.

- Volume — The quantity of generated and stored data. The size of the data determines the value and potential insight, and whether it can be considered big data or not.
- Variety — The type and nature of the data. This helps people who analyze it to effectively use the resulting insight. Big data draws from text, images, audio, video; plus it completes missing pieces through data fusion.
- Velocity — In this context, the speed at which the data is generated and processed to

meet the demands and challenges that lie in the path of growth and development. Big data is often available in real-time.

- Veracity — The data must have quality and produce credible results that enable right action when it comes to end of life decision making.
- Value — This concept refers to the ability of turning data into value. It is important to make a case for any attempt to collect big data. It is easy to fall into the buzz trap when embarking on big data initiatives without a clear understanding of the value it will bring.

3. Big Data Architecture

Big data architecture is the logical and/or physical layout / structure of how data will be stored, accessed and managed within a big data or IT environment. It logically defines how the big data solution will work, how the core components (hardware, database, software, storage) used, information flow, security, and more. Big data architecture primarily serves as the key design reference for big data infrastructures and solutions. Also it entails interconnecting and organizing existing resources to serve big data needs.

Big data architecture is created by big data designers/architects before physically implementing a big data solution. Creating big data architecture generally requires the understanding the business/organization and the need of big data. Typically, big data architecture consists of four different logical layers:

- Big data sources: Think in terms of all of the data available for analysis, coming in from all channels. Clarify what data is required to perform the kind of analyses as needed. The data will vary in format (structured data, semi-structured data, or unstructured data) and origin.
- Data messaging and store layer: This layer is responsible for acquiring data from the data sources, if necessary, converting it to a format that suits how the data is to be analyzed. Compliance regulations and governance policies in different platforms dictate the appropriate storage for different types of data. For example, Hadoop Distributed File System (HDFS) and Relational Database Management System (RDBMS) warehouse are different big data platforms, the transferred data between them might need to be converted so it can be stored and further processed.
- Analysis layer: The analysis layer reads the data digested by the data messaging and store layer. In some cases, the analysis layer accesses the data directly from the data source. Designing the analysis layer requires careful forethought and planning. Decisions must be made with regard to how to manage the tasks to: (1) Produce the desired analytics; (2) Derive insight from the data; (3) Find the entities required; (4) Locate the data sources that can provide data for these entities; (5) Understand what algorithms and tools are required to perform the analytics.
- Consumption layer: This layer consumes the output provided by the analysis layer.

The consumers can be visualization applications, human beings, business processes, or services.

4. Management Tools

The architecture of big data must be synchronized with the support infrastructure of the organization. To date, all of the data which are used by organizations are stagnant. Data is increasingly sourced from various fields that are disorganized and messy, such as information from machines or sensors and large sources of public and private data. Previously, most companies were unable to either capture or store these data, and available tools could not manage the data in a reasonable amount of time. However, the new big data technology improves performance, facilitates innovation in the products and services of business models, and provides decision-making support. Big data technology aims to minimize hardware and process costs and to verify the value of big data before committing significant company resources. Properly managed big data are accessible, reliable, secure, and manageable. Here, we briefly discuss data management tools.

With the evolution of computing technology, immense volumes can be managed without requiring supercomputers and high cost. Many tools and techniques are available for data management, including Google BigTable, Simple DB, Not Only SQL (NoSQL), Data Stream Management System (DSMS), MemcacheDB, and Voldemort. However, companies must develop special tools and technologies that can store, access, and analyze large amounts of data in near-real time because big data differs from the traditional data and cannot be stored in a single machine. Furthermore, big data lacks the structure of traditional data. For big data, some of the most commonly used tools and techniques are Hadoop, MapReduce, and Big Table. These innovations have redefined data management because they effectively process large amounts of data efficiently, cost-effectively, and in a timely manner.

5. Examples of Application

The growth of data has affected across practically every industry, whether it is business sector or the world of science. Here, we present several typical examples of the industries can benefit from big data application.

- Banking

With large amounts of information streaming in from countless sources, banks are faced with finding new and innovative ways to manage big data. While it’s important to understand customers and boost their satisfaction, it’s equally important to minimize risk and fraud while maintaining regulatory compliance. Big data brings big insights, but it also requires financial institutions to stay one step ahead of the game with advanced analytics.

- Education

Educators armed with data-driven insight can make a significant impact on school systems, students and curriculums. By analyzing big data, they can identify at-risk students, make sure

students are making adequate progress, and can implement a better system for evaluation and support of teachers and principals.

- Government

When government agencies are able to harness and apply analytics to their big data, they gain significant ground when it comes to managing utilities, running agencies, dealing with traffic congestion or preventing crime. But while there are many advantages to big data, governments must also address issues of transparency and privacy.

- Health Care

Patient records. Treatment plans. Prescription information. When it comes to health care, everything needs to be done quickly, accurately and, in some cases, with enough transparency to satisfy stringent industry regulations. When big data is managed effectively, health care providers can uncover hidden insights that improve patient care.

- Manufacturing

Armed with insight that big data can provide, manufacturers can boost quality and output while minimizing waste–processes which are key in today's highly competitive market. More and more manufacturers are working in an analytics-based culture, which means they can solve problems faster and make more agile business decisions.

21.3　Cloud Computing

1. Definition of Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. More specifically, cloud computing is an outsourcing of computer programs. Using cloud computing, users are able to access software and applications wherever they are; the computer programs are being hosted by an outside party and reside in the cloud. This means that users do not have to worry about things such as storage and power, they can simply enjoy the end result. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a public utility.

Since the launch of Amazon Elastic Compute Cloud (EC2) in 2006, the availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing has led to growth in cloud computing.

2. Deployment Models

The type of deployment model refers to where the cloud computing service is owned by a

provider. The provider offers the highest level of efficiency to the users in a shared but secure environment. In cloud computing-speak, the firm using the service is referred to as a tenant in a cloud. Here, we cover the four cloud deployment models which are defined by National Institute of Standards and Technology (NIST). Each deployment model has its own benefits, disadvantages, and value proposition.

- Public cloud. In this case, the cloud infrastructure is available to the general public or a large industry group. It is typically owned by an organization selling cloud services.
- Private cloud. Private cloud and internal cloud describe offerings deploying cloud computing on private networks. It is still cloud computing, and may be managed by the organization or a third party and may exist on premise or off premise.
- Community cloud. In this case, the cloud infrastructure is shared by a group of organizations called a 'community'. This community has shared concerns such as a mission, security requirements, policy, and compliance considerations. It may be managed by the organizations or a third party and may exist on premise or off premise.
- Hybrid cloud. In the case of a hybrid cloud deployment solution, the cloud infrastructure is a composition of more than 2 clouds (private, community or public) which remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability.

3. Service Models

Cloud computing can be defined as the use of new or existing computing hardware and virtualization technologies to form a shared infrastructure that enables web-based value added services. The three predominant service models are infrastructure, platform, and software as a service

- SaaS (Software as a Service). This service mainly involves licensing of software applications to the users via the internet. It's basically an on-demand service where users are charged on a subscription basis for the software applications, thereby providing access to cloud-based apps through the internet. It's a delivery model offered by third-party providers to the end users over the internet.
- IaaS (Infrastructure as a Service). It's mainly a computing infrastructure where external cloud providers provide the hardware on a pay-per-use basis. They provide everything from virtual private networks to operating systems. In simple terms, they provide the raw materials for IT and the users pay for only the resources they use.
- PaaS (Platform as a Service). It's the most complex layer of cloud computing which shares some resemblance with SaaS, but instead of licensing software to the users, it creates a platform for the developers to deliver hardware and software tools, the basic stuff that's needed for application development.

Technical Notes to the Text

1. cloud computing，云技术。云技术指在广域网或局域网内将硬件、软件、网络等系列资源统一起来，实现数据的计算、储存、处理和共享的一种托管技术。

2. information technology，信息技术。信息技术指主要用于管理和处理信息所采用的各种技术的总称。它主要是应用计算机科学和通信技术来设计、开发、安装和实施信息系统及应用软件，主要包括传感技术、计算机与智能技术、通信技术和控制技术。

3. server，服务器。服务器指提供计算服务的设备。由于服务器需要响应服务请求，并进行处理，因此，一般来说，服务器应具备承担服务并且保障服务的能力。

4. social media，社会化媒体。社会化媒体是通过人与人之间的联系而产生出的新媒体形式，每个人或组织在社会网络中，同时扮演着信息传播者与接收者的角色。

5. data fusion，数据融合。数据融合指利用计算机技术将来自多个传感器或多源的观测信息进行分析、综合处理，从而得出决策和估计任务所需的信息的处理过程。

6. logical layer，逻辑层。在大数据架构领域中，它提供了组织执行特定功能的软件的方法。

7. information flow，信息流。从现代信息技术研究、发展、应用的角度来定义，信息流指的是信息处理过程中信息在计算机系统和通信网络中的流动。

8. structured data，结构化的数据。它指可以使用关系型数据库表示和存储数据，表现为二维形式的数据。

9. semi-structured data，半结构化数据。它属于结构化数据的一种形式，并不符合关系型数据库或用其他数据表的形式关联起来的数据模型结构，但包含相关标记，用来分隔语义元素及对记录和字段进行分层。因此，它也被称为自描述的结构。

10. unstructured data，非结构化数据。它指没有固定结构的数据，如各种文档、图片、视频/音频等都属于非结构化数据。

11. virtualization technology，虚拟化技术。它指将单个计算机软件环境分割为多个独立分区，每个分区均可以按照需要模拟计算机的一项技术。它的技术实质是通过中间层次实现计算资源的管理和再分配，使资源利用实现最大化。

12. amazon elastic compute cloud，亚马逊弹性计算云。它指让使用者可以租用云端计算机运行所需应用的系统。它借由提供 Web 服务的方式让使用者可以弹性地运行自己的 Amazon 机器映像档，使用者可以在这个虚拟机器上运行任何自己想要的软件或应用程序，提供可调整的云计算能力。它旨在使开发者的网络规模计算变得更为容易。

13. public cloud，公共云。它是基于标准云计算的一个模式，在其中，服务供应商创造资源，如应用和存储，公众可以通过网络获取这些资源。

14. private cloud，私有云。其是为一个客户单独使用而构建的，因而提供对数据、安全性和服务质量的最有效控制。该公司拥有基础设施，并可以控制在此基础设施上部署应用程序的方式。私有云可以部署在企业数据中心的防火墙内，也可以部署在一个安全的主机托管场所。私有云的核心属性是专有资源。

15. community cloud，社区云。其是大的“公有云”范畴内的一个组成部分，指在一定的地域范围内，由云计算服务提供商统一提供计算资源、网络资源、软件和服务能力所形成的云计算形式。即基于社区内的网络互联优势和技术易于整合等特点，通过对区域内各种计算能力进行统一服务形式的整合，结合社区内的用户需求共性，实现面向区域用户需求的云计算服务模式。

16. hybrid cloud，混合云。混合云融合了公有云和私有云，是近年来云计算的主要模式和发展方向。我们已经知道私有云主要是面向企业用户，出于安全考虑，企业更愿意将数据存放在私有云中，但是同时又希望可以获得公有云的计算资源，在这种情况下，混合云被越来越多地采用，它将公有云和私有云进行混合和匹配，以获得最佳的效果，这种个性化的解决方案，达到了既省钱又安全的目的。

17. software as a service，软件即服务。其是指厂商通过互联网提供软件的模式，将应用软件统一部署在自己的服务器上，客户可以根据自己的实际需求，通过互联网向厂商定购所需的应用软件服务，按定购的服务多少和时间长短向厂商支付费用，并通过互联网获得厂商提供的服务。用户不用再购买软件，而改用向提供商租用基于网络的软件来管理企业经营活动，而且无须对软件进行维护，服务提供商会全权管理和维护软件，软件厂商在向客户提供互联网应用的同时，也提供软件的离线操作和本地数据存储，让用户随时随地都可以使用其定购的软件和服务。对于许多小型企业来说，软件即服务是采用先进技术的最好途径，它消除了企业购买、构建和维护基础设施和应用程序的需要。

18. infrastructure as a service，基础设施即服务。消费者通过互联网可以从完善的计算机基础设施获得服务，这类服务称为基础设施即服务。

19. platform as a service，平台即服务。它是指将一个完整的软件研发和部署平台，包括应用设计、应用开发、应用测试和应用托管，都作为一种服务提供给客户。在这种服务模式中，客户不需要购买硬件和软件，只需要利用平台即服务，就能够创建、测试和部署应用和服务。与基于数据中心的平台进行软件开发和部署相比，采用平台即服务的成本和费用要低得多。

Word Bank to the Text

A. Useful new words

virtually	*adv.* 无形，无形中，实际上，事实上，几乎
encompass	*v.* 围绕，包含或包括某事物，完成
overwhelming	*adj.* 势不可挡的，压倒一切的，巨大的
keen	*adj.* 锐利的，厉害的，强烈的，敏锐的，热心的
prospective	*adj.* 未来的，预期的，可能的，有希望的
degradation	*n.* 恶化，堕落，潦倒，毁坏
infrastructure	*n.* 基础设施，基础建设
utilize	*v.* 利用，使用

scalable	*adj.* 可攀登的，可升级的
accommodating	*adj.* 随和的，善于适应的，调节性的
deploy	*v.* 使展开，施展，有效地利用
velocity	*n.* 速率，速度，周转率，高速，快速
veracity	*n.* 诚实，真实
architecture	*n.* 体系结构，建筑学，(总体、层次)结构
compliance	*n.* 服从，听从，承诺，柔软度，顺度
governance	*n.* 管理，统治，支配，统治方式
forethought	*n.* 事先的考虑，远见卓识
transparency	*n.* 透明度，透明，透明性，透明的东西
convert	*v.* (使)转变，兑换，换算，侵占
ubiquitous	*adj.* 无所不在的，普遍存在的
provision	*v.* 为……提供所需物品
tenant	*n.* 房客，佃户
stagnant	*adj.* 污浊的，停滞的，迟钝的，呆笨的

B. Useful expressions

all sorts of	各种各样的
rather than	……而不……，与其……倒不如……
be expected to	被期待做某事，有望做某事，期许
tend to	趋向于，往往，往往会，倾向于，趋向
deal with	应付，对待，惠顾，与……交易
draw from	从……得到，从……召，从(某处)取出
in this context	关于这点，由此而论，由于这个原因，在这个背景下
lie in	躺在，在于，位于
turn into	成为，(使)变成，译成
think in terms of	考虑
be responsible for	为……负责，形成……的原因，主管
benefit from	受益，通过……获益，得力，受用
if necessary	如必要的话
when it comes to	当提到……
in some cases	在某些情况下，有时候
in from	来自
be able to	能够，能，会
rely on	依靠，依赖，信赖，信任
embark on	着手，开始做某事
fall into	堕入，掉进，陷入，陷于
refer to	参考，指的是，涉及，适用于

in this case	既然这样
be defined as	规定为，被称为
to date	迄今，至今，直到今天，目前为止

C. Technical terms and proper names

big data	大数据
cloud computing	云技术
information technology	信息技术
server	服务器
social media	社会化媒体
data fusion	数据融合
logical layer	逻辑层
information flow	信息流
structured data	结构化数据
semi-structured data	半结构化数据
unstructured data	非结构化数据
public cloud	公共云
private cloud	私有云
community cloud	社区云
hybrid cloud	混合云
software as a service	软件即服务
infrastructure as a service	基础设施即服务
platform as a service	平台即服务
umbrella term	涵盖性术语

Exercises

Comprehension of the Text

Ⅰ. Fill in the following blanks.

1. __________ is refer to huge data sets that cannot be processed by a traditional data-processing application software in a single computer.

2. __________ is available for the general public or a large industry group.

3. The three predominant service models in cloud computing are __________ , __________, __________ and __________.

4. The data could be formatted as __________, __________, or __________.

5. List four deployment models of cloud computing: __________, __________, __________, and __________.

Ⅱ. Read the following statements carefully, and decide whether they are true (T) or false (F) according to the text.

1. In big data, it is how much data that goes into the system matters most. (　　)
2. Consumption layer provides input to analysis layer. (　　)
3. SaaS is service provides software applications to the users via the internet. (　　)
4. In cloud computing-speak, the provider offering cloud computing service is referred to as a tenant in a cloud. (　　)
5. The cloud taking data from various resources is inconvertible if the formats are different. (　　)

Ⅲ. Match each of the following terms with the appropriate definition.

server	community cloud
logical layer	IaaS

1. ________ A structuring mechanism for the elements that make up the software solution.
2. ________ A cloud infrastructure is shared by a group of organizations.
3. ________ The delivery of computer infrastructure as a service.
4. ________ A computer program or a device that provides functionality for other programs or devices, called "clients".

Ⅳ. Translate the following into Chinese.

1. analysis layer　　2. information privacy　　3. hidden pattern
4. governance policy　　5. deployment model

Vocabulary

Ⅴ. Fill in the blanks with the words given below. Change the form where necessary.

encompass	prospective	accommodating
scalable	overwhelming	deploy
ubiquitous	utilize	convert

1. Big data helps businesses identify ______ customers and increase sales.
2. Hybrid cloud allows institutions to build and ______ hybrid applications anywhere.
3. Cloud computing and grid computing are ______.
4. The investigations on big data ______ many areas of research, including parallel and distributed computing, database management and software engineering.
5. This analytic technique is very ______ to your data flow.
6. The business data is ________ and distracting—throwing up barriers to productive decision-making.
7. Small companies ______ the power of the cloud through web application.
8. Mobile terminals could be considered as ______ sensors that allow the observation of

human movements on large scale.

Collocation

Ⅵ. Look at the following sentences taken from the text. Just think about what else you can “reduce” and fill in the sentences with the right word.

Example: By applying advanced analytics to big data, banks can **reduce fraud** while maintaining regulatory compliance.

1. These treatment reduce the ri______ of heart disease.
2. We must reduce the dan______ of war by controlling nuclear, chemical and conventional arms.
3. The seller reduce the pr______ of books so as to bring them within the reach of all students.
4. We should avoid using complex algorithm because it can have a performance impact and reduce sp______ of the program.
5. Plans and implements process improvement activities to increase production efficiency and reduce co______ of production.
6. Lean manufacturing is an example of modern production and the only way to reduce wa______.

Summary of the Text

Ⅶ. Choose the best one of the four choices given to fill in each blank.

This chapter introduces background, big data and cloud computing. Nowadays, two __1__ technologies entering into information technology are big data and cloud computing. __2__ these two are not the same; as big data represents content __3__ cloud computing is infrastructure, the __4__ of both the technologies can __5__ excellent results in collecting, analyzing, storing, and using significant amounts of data. To __6__ the phenomenon of big data, industry characterize it as 5 Vs: Volume, Velocity, Variety, Veracity and Value. Big data architecture __7__ uses four logical layers (big data sources, data massaging and store layer, analysis layer and consumption layer) to define the big data solution of how data will be stored, accessed and managed. Many management tools have been developed to manage big data. The most commonly used tools and techniques are Hadoop, MapReduce, and Big Table. Many industries can __8__ big data application, such as banking, education, government, health care, manufacturing and so on. In cloud computing, a cloud can be private, public, community and hybrid. And the three __9__ service models are infrastructure, platform, and software as a service. Cloud users could choose the most satisfactory service combination __10__ their demand.

1. A. sensible	B. ingenious	C. noticeable	D. adaptable
2. A. Because	B. Although	C. Also	D. But
3. A. so	B. but	C. whereas	D. or

4. A. concurrence	B. combination	C. cooperation	D. competition
5. A. receive	B. yield	C. load	D. depict
6. A. describing	B. described	C. describe	D. description
7. A. normally	B. typically	C. currently	D. eventually
8. A. apart from	B. gather from	C. result from	D. benefit from
9. A. precious	B. unconsidered	C. predominant	D. conspicuous
10. A. according to	B. accordingly to	C. accord to	D. accord

Translation

Ⅷ. Translate the following into Chinese.

1. Big data helps large organizations to manage the databases in more advanced and parallel way within the defined time.

2. Cloud computing providers are capable based on investments in their infrastructure, to develop a big data usable and friendly environment and maintain these.

3. The whole world is on a way to have an online business and because of that cloud server has become more advanced with different technology platforms to deploy services online.

4. The big data analytics lifecycle starts with raw data collection and moves to data analytics and decision making.

5. Companies or individuals engaging in cloud computing do not own the physical infrastructure hosting the software platform. Instead, they avoid capital expenditure by renting usage from a third-party provider.

《计算机英语 (第 4 版)》教师用书申请卡

尊敬的老师：

您好！

感谢您使用清华大学出版社的教材，欢迎您对我们教材的不足提出宝贵的意见或建议。为配合教学需要，《计算机英语(第 4 版)》配有教师用书，可向使用本教材的教师免费赠送。

申请方式：请在填写下方信息后，拍照发送到 476371891@qq.com 邮箱，我们会将教师用书免费邮寄给您。为便于确认教师身份，请务必加盖院/系公章。

联系电话：010-62784096

______________________大学______________________学院/系___________学期开设的______________课程，采用清华大学出版社出版的由姜同强主编的《计算机英语(第 4 版)》作为教材，任课教师为______________________，学生为_________个班共__________人。

任课教师需要配套的教师用书。

联系地址：

电话：

E-mail：

任课教师(签名)：

(院/系办公室盖章)